

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN MATHÉMATIQUES  
ET INFORMATIQUE APPLIQUÉES

PAR  
JEAN-FRANÇOIS BEAUDOIN

RÉINGÉNIERIE DE L'EXPERTISE D'UN SYSTÈME À BASE DE  
CONNAISSANCES POUR LE DIAGNOSTIC DE LA PERFORMANCE  
D'ENTREPRISE

DÉCEMBRE 2005

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## SOMMAIRE

Ce mémoire concerne le projet de maîtrise que nous avons développé au cours des deux dernières années. Ce projet a été conduit en étroite collaboration avec le Laboratoire de recherche sur la performance des entreprises [14] (LaRePE), constituante de l'Institut de recherche sur les PME [13] (INRPME), de l'Université du Québec à Trois-Rivières [31] (UQTR). Ce laboratoire a comme première mission le développement d'une expertise sur la performance et la vulnérabilité des entreprises, plus précisément les petites et moyennes entreprises (PME).

En effet, depuis maintenant plusieurs années, le LaRePE s'est développé une large et solide expertise sur la performance d'entreprise, ce qui l'a mené au développement d'outils sur le sujet, dont un outil de diagnostic de la performance d'entreprise, le système PDG. Ce système qui date de 1999 et qui se base sur les techniques de « benchmarking », utilise les données recueillies dans un questionnaire exhaustif préalablement rempli par l'entreprise, afin d'effectuer un diagnostic détaillé sur la performance de cette dernière. Les résultats du diagnostic, qui concernent une panoplie de domaines et de fonctions de l'entreprise, sont regroupés à l'intérieur d'un rapport détaillé sous forme de graphiques de couleurs et de commentaires textuels. De plus, ces commentaires font l'objet de recommandations visant à aider l'entreprise à améliorer ses faiblesses. Ce système fut un réel succès. En effet, plus de 500 entreprises provenant du Canada, des États-Unis et de la France en ont bénéficié.

Malgré son succès, le système PDG possède plusieurs limites au niveau de son implémentation. Plus précisément, il n'a pas été construit avec des techniques d'intelligences artificielles reconnues et les éléments d'expertise sont dispersés dans son code, ce qui nuit à sa flexibilité, limite sa capacité d'adaptation et d'amélioration, et le rend difficile de compréhension. D'un point de vue externe, le système PDG agit comme un système expert. Par contre, d'un point de vue interne, on ne peut le qualifier de système expert au vrai sens du terme de l'intelligence artificielle.

Le projet de maîtrise a donc consisté au développement d'un **tout nouveau système expert PDGII**, construit selon les règles de l'art de l'intelligence artificielle. Pour ce faire, un important travail de **réingénierie et de formalisation des connaissances** fut effectué pour la construction et l'utilisation d'une **base de connaissances** sur le domaine de la performance d'entreprise. Un **engin d'inférence** a aussi été joint au nouveau système pour permettre le raisonnement et le diagnostic de performance. De plus, l'utilisation d'un nouvel **entrepôt de données** ainsi que la construction d'un **nouveau programme utilisateur** sont venues grandement améliorer ce nouveau système PDGII.

## ABSTRACT

The master degree work described herein is based on a project that we have developed over the last two years. This project was done in collaboration with the LaRePE [14] (Laboratoire de recherche sur la performance des entreprises), one of the research labs of the Research Institute for SME [13] (Small and Medium-sized Enterprises) of the Université du Québec à Trois-Rivières [31]. The main goal of the LaRePE is the development of expertise on the performance and the vulnerability of SME.

For the last couple of years, the LaRePE has developed valuable expertise on the evaluation of enterprise performance. In particular, this led to the development in 1999 of a computerized diagnosis tool of enterprise performance based on a benchmarking approach: this system is called the PDG system. With the use of numerous data collected from a comprehensive questionnaire filled by an evaluated SME, this system diagnoses the enterprise performance. The diagnosis results, organized in terms of the enterprise's main business activities, are shown in a detailed report with colour graphics and texts. Moreover, this report contains recommendations to help the enterprise improve its performance. This system is a success and has been used by more than 500 enterprises from Canada, USA and France.

However, despite its success and suitability, the PDG system was obsolete from an information technology perspective. Indeed, the software implementation suffered from important weaknesses that rendered its functioning, updating, understanding, and evolution rather difficult. More specifically, the PDG system was not built with established artificial intelligence techniques and, amongst its most serious problems, the system's expertise elements were scattered throughout the implementation. As a result of this situation, although the system could be viewed as an expert system from the outside, it could not from the inside, thus negatively impacting its true potential as a knowledge-based system.

This master degree project has consisted in the development of a **brand new PDGII expert system** using an up-to-date artificial intelligence foundation. An important **reengineering and formalization of knowledge** was carried out and that led to the development and the use of a **knowledge base** on the enterprise performance domain. Also, an **inference engine** was added to the PDGII system for the reasoning and diagnosis capabilities. Finally, the use of a new **data warehouse** and the implementation of a new **user program** have improved significantly the overall system.

## REMERCIEMENTS

En tout premier lieu, je remercie **mes parents et ma sœur**, pour leur soutien et encouragement constant, tout au long du cheminement de mes études et de ce projet de maîtrise. Merci d'être là pour moi. C'est grâce à vous et votre amour que je suis la personne que je suis devenue et qui me permet de foncer dans la vie. Un énorme merci.

Merci à **Sylvain Delisle** pour avoir dirigé mon projet de maîtrise et pour m'avoir accompagné durant tout le processus. Son appui, sa persévérance, son implication et sa grande disponibilité sont tous des facteurs sans lesquels la réalisation de ce projet n'aurait pu avoir lieu. Merci beaucoup.

Merci à **Mathieu Dugré**, qui a contribué à plusieurs éléments du système PDGII. Son développement de l'entrepôt de données, ainsi que son travail important dans la préparation des données et dans la conception du programme utilisateur (et du rapport) du nouveau système PDGII, ont permis sans aucune doute la réussite de ce projet.

Je remercie tout particulièrement madame **Josée St-Pierre**, directrice du LaRePE, pour m'avoir permis d'être partie intégrante de ce grand projet. Merci à tous les membres de **l'équipe du LaRePE** pour leur contribution de près ou de loin dans ce projet.

Finalement, je tiens à remercier personnellement **parents et ami(e)s**, pour leurs nombreux et constants encouragements qui me furent d'un très grand soutien durant ces années d'études.

***Merci beaucoup!***

# TABLE DES MATIÈRES

<b>SOMMAIRE .....</b>	<b>I</b>
<b>ABSTRACT .....</b>	<b>II</b>
<b>REMERCIEMENTS .....</b>	<b>III</b>
<b>TABLE DES MATIÈRES .....</b>	<b>IV</b>
<b>LISTE DES FIGURES .....</b>	<b>VII</b>
<b>LISTE DES TABLEAUX.....</b>	<b>IX</b>
<b>LISTE DES ABBRÉVIATIONS.....</b>	<b>X</b>
<b>CHAPITRE 1 INTRODUCTION.....</b>	<b>1</b>
1.1 Processus de prise de décision .....	1
1.2 Système d'aide à la décision et système expert .....	3
1.3 Contexte .....	3
1.4 Objectif : création d'un nouveau système .....	4
1.5 Organisation du document .....	5
<b>CHAPITRE 2 LA PROBLÉMATIQUE .....</b>	<b>7</b>
2.1 Le système utilisé par le laboratoire .....	7
2.1.1 Présentation du système .....	7
2.1.2 Limites du système.....	14
2.2 Objectifs visés .....	18
<b>CHAPITRE 3 ÉTAT DE L'ART.....</b>	<b>19</b>
3.1 Système d'aide à la décision .....	19
3.1.1 Définition .....	19
3.1.2 Caractéristiques .....	19
3.1.3 Composition .....	21
3.1.4 Domaines et exemples d'application .....	23
3.2 Système expert .....	25
3.2.1 Définition .....	25
3.2.2 Caractéristiques .....	26
3.2.3 Composition .....	26
3.2.4 Domaines et exemples d'application .....	28
3.2.5 Système d'aide à la décision versus système expert .....	29
3.3 Base de connaissances.....	29

3.3.1 Définition .....	29
3.3.2 Niveaux de connaissances.....	29
3.3.3 Catégories de connaissances .....	30
3.3.4 Types de connaissances .....	31
3.3.5 Ingénierie de la connaissance.....	32
3.3.5.1 Acquisition des connaissances .....	33
3.3.5.2 Validation des connaissances .....	35
3.3.5.3 Représentation et saisie des connaissances .....	36
3.3.5.4 Inférence.....	38
3.3.5.5 Explication et justification .....	41
<b>CHAPITRE 4 RÉINGÉNIERIE DES CONNAISSANCES ET DÉVELOPPEMENT D'UN NOUVEAU SYSTÈME .....</b>	<b>43</b>
4.1 Présentation globale du système .....	43
4.2 Éléments de solutions du nouveau système .....	45
4.2.1 La base de connaissances .....	45
4.2.1.1 Conception et technologie.....	46
4.2.1.2 Contenu .....	58
4.2.1.3 Solutions apportées .....	73
4.2.2 Programme utilisateur .....	74
4.2.2.1 Technologie et présentation du programme.....	74
4.2.2.2 Solutions apportées .....	87
4.2.3 Entrepôt de données .....	87
4.2.3.1 Conception et technologie.....	87
4.2.3.2 Solutions apportées .....	87
4.2.4 Le rapport de diagnostic.....	88
4.2.4.1 Présentation du rapport .....	88
4.2.4.2 Conception et technologie.....	92
4.2.4.3 Solutions apportées .....	94
<b>CHAPITRE 5 TESTS, VALIDATION, EXEMPLE .....</b>	<b>96</b>
5.1 Tests et validation .....	96
5.2 Tests et validation des améliorations futures .....	103
5.3 Processus complet d'un diagnostic PDGII.....	105
<b>CHAPITRE 6 AMÉLIORATIONS POUR LE FUTUR .....</b>	<b>108</b>
6.1 Traitement de l'incertitude .....	108
6.1.1 Mise à jour Bayésienne (« <i>Bayesian updating</i> »).....	110
6.1.1.1 La mise à jour Bayésienne et le système PDGII .....	112
6.1.2 Facteurs de certitude (« <i>Certainty factors</i> ») .....	113
6.1.2.1 Les facteurs de certitude et le système PDGII .....	113
6.1.3 Logique floue (« <i>Fuzzy logic</i> ») .....	114
6.1.3.1 La logique floue et le système PDGII .....	116
6.2 Ontologie.....	119

<b>CHAPITRE 7 CONCLUSION .....</b>	<b>127</b>
<b>RÉFÉRENCES.....</b>	<b>129</b>



## LISTE DES FIGURES

Figure 2.1 Rapport de diagnostic PDG : page des ressources humaines .....	11
Figure 2.2 Rapport de diagnostic : performance globale et suggestions.....	12
Figure 2.3 Processus de production d'un diagnostic PDG et du rapport [8].....	13
Figure 2.4 Modules du système PDG .....	15
Figure 3.1 Les 4 styles de décision [21].....	20
Figure 3.2 Architecture habituelle d'un DSS [30] .....	22
Figure 3.3 Vulgarisation en schéma d'un système expert.....	25
Figure 3.4 Architecture habituelle d'un système expert [20].....	27
Figure 3.5 Différents types de connaissances [30].....	31
Figure 3.6 Processus d'ingénierie de la connaissance [30].....	33
Figure 3.7 Méthode manuelle d'acquisition des connaissances [30].....	34
Figure 3.8 Critères de validation des connaissances [30] .....	36
Figure 3.9 Exemple de hiérarchie de « frames » représentant des véhicules [30] .....	38
Figure 3.10 Exemple d'un arbre d'inférence [30].....	40
Figure 4.1 Architecture du système PDGII.....	45
Figure 4.2 Architecture de la base de connaissances .....	55
Figure 4.3 Formalisation en 3 niveaux des connaissances du système PDGII.....	57
Figure 4.4 Partie générique de la base de connaissances (banque de critères) .....	60
Figure 4.5 Choix des paramètres de base.....	76
Figure 4.6 Choix du profil de diagnostic .....	77
Figure 4.7 Choix de l'entreprise.....	78
Figure 4.8 Choix de la devise et du groupe témoin .....	79
Figure 4.9 Modification d'expertise : pondération des critères d'évaluation .....	80
Figure 4.10 Présentation d'un résumé des paramètres du diagnostic .....	81
Figure 4.11 Production du diagnostic PDGII par le système.....	82
Figure 4.12 Configuration du rapport : modification d'un graphique .....	83
Figure 4.13 Configuration du rapport : modification d'une recommandation.....	84
Figure 4.14 Liste des diagnostics déjà produits d'un utilisateur.....	86
Figure 4.15 Rapport de diagnostic : portrait global de la performance de l'entreprise ...	89
Figure 4.16 Rapport de diagnostic : portrait du groupe témoin .....	90
Figure 4.17 Rapport de diagnostic : performance de l'entreprise sur un secteur donné..	91
Figure 4.18 Exemple d'un « block » dans un PDF .....	93
Figure 4.19 Configuration d'un « block ».....	94
Figure 5.1 Processus itératif des tests .....	97
Figure 5.2 Processus de maintenance de la base de connaissances .....	104
Figure 5.3 Exemple d'une page de résultats du rapport PDGII. ....	106
Figure 5.4 Exemple d'une page de résultats du rapport PDGII. ....	107
Figure 6.1 Exemple des 3 étapes de logique floue avec Flint. [17] .....	118
Figure 6.2 Composition d'une ontologie complète.....	119
Figure 6.3 Dictionnaire de variables : recherche de variables .....	121

Figure 6.4 Dictionnaire de variables : résultat d'une recherche de variables .....	121
Figure 6.5 Informations pour les statistiques sur une variable.....	122
Figure 6.6 Informations supplémentaires sur une variable de questionnaire.....	122
Figure 6.7 Informations supplémentaires sur une variable calculée .....	123
Figure 6.8 Composition de l'ontologie avec le dictionnaire de variables.....	124

## LISTE DES TABLEAUX

Tableau 2.1 Fonctions et sections d'intégration du diagnostic PDG. ....	7
Tableau 4.1 Description du frame d'un critère d'évaluation et du frame « benchmark_data ». ....	59
Tableau 4.2 Description du frame « global » . ....	62
Tableau 4.3 Description du frame « profil » . ....	63
Tableau 4.4 Description du frame « section » . ....	64
Tableau 4.5 Description du frame « critere_PDGI » . ....	66
Tableau 4.6 Description du frame « critere_selectionne » . ....	68
Tableau 4.7 Description du frame jointure_criteres_selectionnes » . ....	69
Tableau 4.8 Description des frames « commentaire_critere_pratique » et « commentaire_critere_resultat » . ....	71
Tableau 4.9 Description des étapes et règles d'un diagnostic.....	73
Tableau 5.1 Test de comparaison entre les résultats du système PDG et ceux du nouveau système PDGI. ....	101
Tableau 6.1 Détails des informations sur les variables du dictionnaire de variables.....	125

## LISTE DES ABBRÉVIATIONS

LaRePE	Laboratoire de recherche sur la performance des entreprises
INRPME	Institut de recherche sur les PME
UQTR	Université du Québec à Trois-Rivières
DSS	« Decision Support System »
SIAD	Système d'aide à la décision
LPA	Logic Programming Associates Ltd.
PME	Petite et moyenne entreprise
PDG	Système PDG® (Performance, Développement, Gestion)
PDGII	Système PDGII (Performance, Développement, Gestion)
JAVA	Langage de programmation JAVA
DLL	« <i>Dynamic Link Library</i> »
JSP	Langage de programmation Java Server Page
VB	Langage de programmation Visual Basic
PDF	« Portable Document Format ».

# CHAPITRE 1

## INTRODUCTION

Chaque jour, l'humain fait face à des situations où il doit prendre des décisions. Ces dernières peuvent avoir des répercussions plus ou moins importantes selon le contexte et les raisons pour lesquels elles ont été prises. Le contexte de gestion d'une entreprise en fait partie. Une bonne prise de décision en entreprise requiert bien souvent l'analyse d'un grand nombre de données. Cette analyse est un processus qui peut être long et complexe selon le nombre de données à prendre en considération et peut avoir des répercussions sur la bonne performance future d'une entreprise.

Ce chapitre introduit le travail de maîtrise sur lequel j'ai travaillé durant les 2 dernières années. Il a été fait dans le contexte du Laboratoire de recherche sur la performance des entreprises [14] (LaRePE), constituante de l'Institut de recherche sur les PME [13] (INRPME), de l'Université du Québec à Trois-Rivières [31]. Ce laboratoire effectue des recherches sur la performance des entreprises, en particulier les petites et moyennes entreprises (PME), et en fait le développement d'une expertise. Comme nous le verrons dans ce document, l'informatique et l'utilisation de l'intelligence artificielle dans le contexte du LaRePE peuvent être d'un grand secours pour aider les dirigeants d'une entreprise à prendre de bonnes décisions rapidement et ainsi à améliorer la performance de leur entreprise.

### 1.1 Processus de prise de décision

Le processus de prise de décision est un élément très important pour la bonne performance d'une entreprise. Ce processus se divise en 4 phases [30] :

- *La phase d'intelligence.* C'est dans cette phase que la réalité est examinée et que le problème éventuel est détecté et défini.
- *La phase du design.* C'est durant cette phase qu'un modèle sera construit. Ce dernier doit être le plus possible fidèle à la réalité. Une fois ce modèle construit, il doit être validé pour, par la suite, être utilisé pour identifier les différentes solutions possibles pour résoudre le problème.
- *La phase du choix.* C'est dans cette phase que le choix de la bonne solution sera fait parmi toutes celles identifiées lors de la phase précédente. Une fois ce choix fait, il sera testé en détail pour s'assurer que la solution est réalisable.

- *La phase de l'implémentation.* Une fois la solution choisie et testée, on doit l'appliquer pour résoudre le problème. La phase de l'implémentation est un succès si le problème est résolu. Si ce n'est pas le cas, il faut reprendre le processus de prise de décision à une des phases précédentes, selon le cas.

Des problèmes d'efficacité existent concernant ces phases du processus de prise de décision quand il est fait par un humain. Tout d'abord, la phase d'intelligence, dans laquelle on détecte les problèmes éventuels, peut s'avérer inefficace pour un humain. En effet, la détection et le diagnostic d'un problème est quelque chose d'assez pointu. Bien souvent, plusieurs paramètres et plusieurs données doivent être pris en compte pour détecter un problème, ce qui implique que la détection d'un problème est parfois longue. Ou encore pire, l'oubli d'un seul de ses paramètres ou données peut entraîner la non-détection du problème et d'éventuelles difficultés sur la performance de l'entreprise.

D'un autre côté, la phase de design, celle où on modélise la situation réelle et on identifie les différentes solutions aux problèmes détectés, peut aussi s'avérer non efficace. En effet, modéliser la situation réelle peut être, selon le cas, un exercice complexe, encore une fois en raison du grand nombre de paramètres et de données. Il l'est encore plus s'il doit être refait chaque fois que l'on a détecté un problème et que l'on veut trouver des solutions potentielles.

Bref, bien qu'un bon nombre le sont déjà, tous les problèmes dans un processus idéal de prise de décision devraient être détectés rapidement, afin d'éviter des effets nuisibles à l'entreprise. De plus, une fois la détection d'un problème potentiel, la prise de décision devrait se faire rapidement, encore une fois pour des raisons d'efficacité de la performance de l'entreprise. Les dirigeants d'entreprise peuvent être tentés de prendre des décisions rapides lors de la détection d'un problème, mais comme on l'a vu précédemment, une décision prise trop rapidement risque d'avoir été mal évaluée et d'être inefficace. À l'opposé, comme on vient aussi tout juste de le voir, bien évaluer une décision peut être trop long en raison du grand nombre de paramètres et de données, ce qui sera encore une fois inefficace pour l'entreprise. Il est donc clair qu'une prise de décision idéale et efficace doit être à la fois bien évaluée et rapide.

Ce concept de prise de décision est à la base de ce mémoire. Comme nous le verrons plus loin, il est possible de créer des systèmes informatiques afin d'aider les preneurs de décisions à détecter les problèmes et à prendre des décisions bien évaluées, éclairées, et ce, de façon rapide. On nomme ces systèmes, systèmes d'aide à la décision ou systèmes experts, selon leur application et leur but d'utilisation.

## 1.2 Système d'aide à la décision et système expert

Il existe ce qu'on appelle des systèmes informatiques d'aide à la décision (SIAD) ou « *decision support systems* » (DSS). Comme la littérature du domaine utilise en grande majorité cette dernière abréviation, nous l'utiliserons dorénavant dans ce document.

Un DSS, comme son nom le définit, est un système informatique permettant à son utilisateur de prendre des décisions plus éclairées et plus sûres sur un domaine en particulier. Il est bien important de comprendre que ce n'est pas un système qui prendra une décision, dans le but de l'appliquer de manière automatique. Au contraire, ce type de système peut faire l'analyse de problèmes et proposer les meilleures solutions possibles, mais laisse toujours le libre choix à son utilisateur de prendre lui-même la décision finale.

D'un autre côté, un système expert (« *knowledge-based expert system* ») est un système qui utilise des connaissances humaines saisies dans un ordinateur afin de résoudre des problèmes normalement résolus par des experts. L'élément central d'un système expert est la base de connaissances, dans laquelle se retrouve toute l'expertise voulue sur le domaine d'application du système expert. Le but ultime d'un système expert est de résoudre un problème plus rapidement et plus efficacement qu'un expert humain. Cette meilleure efficacité du système expert se traduit en autres par la réduction des possibilités d'erreur.

Il peut être difficile de différencier un système expert d'un DSS. En effet, les deux types de systèmes tentent de simuler le raisonnement d'un humain à l'aide de ces connaissances afin d'analyser un problème. Cependant, le système expert est ultimement construit afin de remplacer le preneur de décision et ainsi, de penser et agir à sa place. Cela est contraire à la philosophie d'un DSS qui est d'accompagner le preneur de décision afin qu'il soit en mesure de prendre la meilleure décision possible.

Ces concepts seront définis en détail dans le prochain chapitre de ce document.

## 1.3 Contexte

Tel que mentionné précédemment, le projet à la base de ce mémoire de maîtrise, a été conçu en contexte réel d'application et d'utilisation au sein du Laboratoire de recherche sur la performance des entreprises (LaRePE).

Le LaRePE a comme mission première de développer une expertise unique sur la performance et la vulnérabilité des PME en réalisant des recherches de pointe sur ces sujets. Il a été créé en 1997 pour répondre aux besoins d'un groupe de dirigeants de PME de se doter d'un outil d'évaluation pour la gestion de la performance de leur entreprise.

En collaboration avec le Groupement des chefs d'entreprise du Québec [9] et avec la contribution financière d'organismes gouvernementaux, l'équipe de recherche multidisciplinaire du LaRePE a développé un outil de « benchmarking » répondant aux besoins des chefs d'entreprise : le système PDG® [29]. Ce système, qui a nécessité près de cinq années de recherche et développement, a permis la constitution d'une base de données qui est utilisée aux fins de la recherche sur les PME. Cette base de données est alimentée continuellement par l'ajout d'entreprises et de nouvelles dimensions liées à la performance : celle-ci est composée majoritairement d'entreprises canadiennes, auxquelles s'ajoutent quelques entreprises américaines et, depuis peu, des entreprises françaises.

Le système PDG® est à la base de ce projet de maîtrise. En effet, comme nous le verrons dans le prochain chapitre, cet outil a été conçu afin d'aider les dirigeants d'entreprise à détecter rapidement les problèmes de performance dans leur entreprise et à les aider à prendre de bonnes décisions pour améliorer cette performance. Le système PDG® est un système utile, qui a connu un succès certain, sauf qu'il n'a pas été construit selon les règles de l'art de l'intelligence artificielle. Il ne peut donc pas être qualifié de système expert. Le système PDG® possède par conséquent des limites de conception qui nuisent à son bon fonctionnement, sa compréhension et son évolution. Cette situation, qui constitue la problématique à laquelle ce projet de maîtrise a dû répondre, sera décrite en détail dans le prochain chapitre de ce mémoire.

## **1.4 Objectif : création d'un nouveau système**

L'objectif et la solution ont résidé dans la conception d'un tout nouveau système. Ce nouveau système a le même objectif principal que le système PDG®, soit le diagnostic de la performance d'entreprise. Cependant, ce nouveau système a été construit selon les règles de l'art de l'intelligence artificielle et contient des éléments qu'un DSS et qu'un système expert doit posséder.

Les avantages de la construction d'un nouveau système sont nombreux et seront détaillés dans ce document. Voici ces avantages :

- Représentation explicite des éléments de connaissance (expertise) sur le domaine du diagnostic de la performance d'entreprise.
- Flexibilité du nouveau système, par conséquent : facilité d'entretien, d'amélioration et d'évolution de ce dernier.
- Plus grande facilité d'utilisation.
- Amélioration des performances d'exécution.

Nous avons vu précédemment qu'il y a une différence subtile entre la notion de DSS et de système expert. La grande différence réside dans le fait qu'un DSS aide, suggère et



accompagne le preneur de décision alors qu'un système expert remplace le preneur de décision en prenant, et en appliquant selon le cas, lui-même la décision. Nous pourrions donc dire que le nouveau système, qui diagnostique des faiblesses dans la performance d'une entreprise et qui ensuite propose des solutions, sans prendre la décision, pourrait être qualifié de DSS. Cependant, comme nous l'avons aussi vu, un système expert est un système qui est basé sur les connaissances, et sa grande force réside dans sa base de connaissances qui contient tous les éléments d'expertise qu'un expert possède sur le domaine. Or, comme nous le verrons, l'amélioration majeure du nouveau système consiste dans la conception d'une base de connaissances sur le domaine de la performance d'entreprise. Le système pourrait donc aussi être qualifié de système expert.

Par conséquent, étant donné que ce nouveau système possède les caractéristiques d'un DSS, mais aussi d'un système expert, il s'agit plutôt d'un système expert hybride. Pour simplifier la lecture de ce document, le terme système expert sera dorénavant utilisé pour qualifier le nouveau système de diagnostic de performance d'entreprise. De plus, on appellera le nouveau système : **le système PDGII**.

Il est à noter qu'en attendant l'implantation finale du nouveau système, le système PDG® actuel continuera d'être utilisé par le laboratoire pour le diagnostic de la performance d'entreprise.

## 1.5 Organisation du document

Ce mémoire a donc comme élément central le projet PDGII réalisé durant les deux dernières années au LaRePE.

La problématique entourant le système PDG® déjà existant sera présentée dans le **chapitre 2**. Ce système sera tout d'abord défini et expliqué en détail afin de bien comprendre de quelle manière et dans quel contexte il a été conçu. Son fonctionnement général et son architecture seront aussi décrits. Ces explications seront très utiles afin de mieux saisir et cerner la problématique. En effet, dans ce même chapitre, la problématique, qui tournera essentiellement autour des limites et faiblesses du système PDG®, sera ensuite présentée en détail.

Le **chapitre 3** portera sur une présentation de l'état de l'art des concepts qui concernent le projet. Il y aura recension de la littérature pour des concepts comme les systèmes d'aide à la décision, les systèmes experts, l'ingénierie des connaissances, et les bases de connaissances. Des exemples d'applications réelles de DSS et de systèmes experts seront aussi présentés.

La solution à la problématique sera exposée en détail dans le **chapitre 4**. Nous verrons qu'elle a consisté en la construction d'un tout nouveau système pour le diagnostic de la performance d'entreprise : le système PDGII. Des techniques à jour d'intelligence artificielle ont été utilisées dans ce nouveau système afin de pallier aux différentes limites du système PDG®.

Par la suite, le **chapitre 5** décrira les différents tests, passés et futurs, servant à valider la bonne conformité des résultats produits par le système expert de diagnostic de performance d'entreprise PDGII. Il y aura aussi dans ce chapitre la présentation d'un exemple complet du processus de production d'un diagnostic à l'aide du système PDGII, du questionnaire jusqu'à la production du rapport final des résultats de la performance de l'entreprise.

Le **chapitre 6** sera consacré aux développements futurs entourant le nouveau système. Deux concepts intéressants ont fait l'objet d'une étude sur leur potentiel d'utilisation dans le nouveau système et cette dernière sera présentée dans ce chapitre. Il s'agit des concepts de traitement de l'incertitude des connaissances et du développement d'une ontologie approfondie pour la base de connaissances.

Finalement, le **chapitre 7** sera présenté en guise de conclusion.

## CHAPITRE 2

### LA PROBLÉMATIQUE

#### 2.1 Le système utilisé par le laboratoire

En 1999, l'équipe de recherche multidisciplinaire du LaRePE a développé, en collaboration avec le Groupement des chefs d'entreprise du Québec et avec la contribution financière d'organismes gouvernementaux, un outil répondant aux besoins des chefs d'entreprise, le système PDG [5, 24, 29]. Ce système permet le diagnostic de la performance d'entreprise (PME) et aide les dirigeants d'entreprise à apporter les corrections nécessaires au besoin pour leur entreprise. Le système PDG fut un grand succès, ayant été utilisé par plus de 500 PME canadiennes, américaines, et françaises.

Bien qu'il ait été un succès, ce système, qui tend à vouloir imiter un système expert d'un point de vue externe [6], n'en est pas un au niveau de l'implémentation. En effet, à première vue et de la manière dont il fonctionne, il apparaît comme un système expert, par contre, il n'a pas été construit et implémenté selon les règles de l'art de l'intelligence artificielle. De plus, lorsqu'on évalue le système PDG, on se rend compte de limites dans sa conception qui le nuisent sur plusieurs plans, autant au niveau de son bon fonctionnement, de sa flexibilité, que de sa compréhensibilité.

Présentons maintenant en détail ce système et ses limites pour bien en comprendre la problématique qui l'entoure.

##### 2.1.1 Présentation du système

Le système PDG est un système qui permet de faire l'évaluation d'une entreprise afin de diagnostiquer son potentiel et son niveau de performance. Cette évaluation est faite sur 5 fonctions de l'entreprise, suivie de 2 sections d'intégration :

Fonctions	Sections d'intégration
Ressources humaines	Efficacité
Systèmes de production	Vulnérabilité
Gestion de la production	
Gestion et contrôle	
Ventes et marketing	

Tableau 2.1 Fonctions et sections d'intégration du diagnostic PDG.

Il est à noter que dans ce document, ces fonctions et sections d'intégration, seront présentées comme étant des secteurs d'évaluation ou secteurs d'activités.

L'évaluation du système PDG est produite à l'aide d'une technique très utilisée et connue, nommée « benchmarking » [25, 33]. Cette dernière consiste en l'analyse comparative des données d'une entreprise à celles d'autres entreprises dans le but de diagnostiquer quelque chose. La comparaison peut se faire avec une norme théorique, avec une donnée « idéale », avec une entreprise concurrente, avec une entreprise leader ou bien avec un groupe témoin d'entreprises défini selon différents critères. C'est d'ailleurs cette dernière méthode qui est utilisée dans le cadre du système PDG.

Voici les différents critères de sélection du groupe témoin utilisés dans le système PDG [29] :

- **Ventes :** Les ventes de l'entreprise sont souvent utilisées pour mesurer sa taille. On évite habituellement de comparer des entreprises de taille différente parce qu'elles n'ont pas le même degré de développement, ne bénéficient pas des mêmes opportunités de marché ni de certaines économies d'échelle réservées souvent aux plus grandes. *Les entreprises ayant un chiffre d'affaires relativement faible et un rythme de croissance plutôt modeste peuvent privilégier ce critère. D'un autre côté, il peut être intéressant pour les entrepreneurs dirigeant de petites entreprises où la progression des ventes est rapide d'opter pour un autre critère de sélection.*
- **Nombre d'employés :** Le nombre d'employés total de l'entreprise est une autre mesure de la taille qui n'est pas toujours une image représentant fidèlement son volume d'affaires. Il est influencé par l'utilisation plus ou moins intensive d'équipements de production, de la sous-traitance et de la diversité de la clientèle. On évite habituellement de comparer des entreprises de taille différente parce qu'elles n'ont pas le même degré de développement, ne bénéficient pas des mêmes opportunités de marché ni de certaines économies d'échelle réservées souvent aux plus grandes. *Les entreprises ayant un nombre relativement faible d'employés devraient privilégier ce critère, à moins que le dirigeant souhaite une croissance rapide qui affectera son développement à court terme.*
- **Sous-traitance :** Le pourcentage des ventes obtenu en sous-traitance a un impact sur le développement des systèmes de production, sur les besoins de structurer les activités de gestion et de contrôle; bref, sur l'ensemble des coûts de l'entreprise. L'ampleur de ces effets dépend du pourcentage de contrats obtenu en sous-traitance, de la qualité de la relation entre le donneur d'ordre et l'entreprise et, finalement, de la présence d'ententes à long terme entre les deux parties. *Les entreprises ayant un pourcentage de contrat obtenu en sous-traitance élevée ont avantage à privilégier ce critère. D'un autre côté, celles qui font affaire avec un*

*faible nombre de clients et qui souhaitent diversifier leurs activités devraient privilégier un autre critère pour voir les structures des autres entreprises.*

- **Ratio d'employés de production / nombre d'employés total :** Le pourcentage d'employés de production qu'embauche une entreprise est fonction de son degré d'automatisation et des contrats qu'elle donne en sous-traitance. Il permet ainsi de comparer des entreprises ayant un degré de développement technologique semblable. *Les entreprises fortement ou très faiblement automatisées ou dont une partie des activités est offerte en sous-traitance devraient favoriser ce critère.*
- **Âge :** L'âge de l'entreprise permet d'estimer sa maturité et son degré de développement. Les entreprises développent généralement leurs pratiques de gestion au même rythme que leurs activités de production et que la progression de leur chiffre d'affaires. Il faut donc éviter de comparer des entreprises ayant un âge très différent, ce qui porterait préjudice aux plus jeunes et serait un mauvais indicateur de leur efficacité et de leur performance. *Les jeunes entreprises devraient privilégier ce critère, alors que les entrepreneurs qui souhaitent évaluer ce dont a besoin leur entreprise pour accroître leur chiffre d'affaires ne devraient pas l'utiliser.*
- **Secteur :** Le secteur d'activités définit les systèmes de production, les types de production, ainsi que les matières premières utilisées. Les entreprises des mêmes secteurs ont généralement une structure d'actifs, technologique et de production semblable. Dans plusieurs cas, le secteur détermine également les délais de paiement et de recouvrement des comptes. Il est préférable de comparer entre elles des entreprises évoluant dans le même secteur, quoique l'influence de celui-ci sur les pratiques et résultats peut être moins importante que le degré de maturité, de sous-traitance ou d'exportation.
- **Type de production :** Le type de production définit toute la structure du plancher de production de l'entreprise, ses systèmes ainsi que son développement technologique. *Cela peut être un critère pertinent pour une entreprise évoluant dans un environnement relativement stable.*
- **Taux de croissance des ventes :** Le rythme de croissance des ventes est un indicateur de la vitesse à laquelle croît l'entreprise, mais surtout de l'instabilité de l'environnement dans lequel elle évolue. Les systèmes de production peuvent être plus ou moins au point, les pratiques de gestion ne suivent souvent pas cette évolution, le personnel est peut-être insuffisant, bref, les résultats financiers pourront être très instables tout comme l'environnement de l'entreprise. Il pourrait ainsi être approprié pour l'entrepreneur de tenir compte de son rythme de croissance pour sélectionner les entreprises auxquelles il sera comparé. *Les*

*entreprises en forte croissance et celles qui désirent maintenir un rythme de croissance fixe ou nul peuvent privilégier ce critère.*

- **Exportation :** Les activités d'exportation entraînent des effets plus ou moins importants sur le développement des systèmes de production, sur les délais de production ainsi que sur les résultats financiers dont, notamment, sur les liquidités. Des conditions de marché différentes de ce que l'on trouve sur le plan local, des coûts de production habituellement supérieurs, des délais significativement allongés suggèrent ainsi de tenir compte de ce critère dans la sélection du groupe témoin. *Les entreprises ayant une forte proportion de leurs ventes à l'étranger ou souhaitant développer de façon significative leurs exportations devraient privilégier ce critère.*

Les données de l'entreprise qui sont utilisées pour le diagnostic sont récupérées grâce à un questionnaire exhaustif de 20 pages que l'entreprise doit remplir. Ce questionnaire a été conçu par une équipe de chercheurs multidisciplinaire en stratégie, ressources humaines, système d'information, génie industriel, logistique, marketing, économie et finance.

D'un autre côté, le développement même du système informatique PDG a été fait par des gens issus du monde de la gestion ainsi que par un informaticien. Le logiciel Microsoft Excel ainsi que le langage de programmation Visual Basic (<http://www.microsoft.com/>) furent utilisés pour construire l'outil de diagnostic et de production du rapport. Comme un grand nombre de calculs statistiques étaient requis, dont pour le calcul des données du groupe témoin, le logiciel SAS (<http://www.sas.com/>) était utilisé pour exécuter ces calculs statistiques. Pour leur part, les données sur les entreprises, recueillies par le questionnaire, étaient stockées dans une base de données Oracle.

La grande force et l'appréciation du système PDG par les entreprises viennent du fait que les résultats du diagnostic de la performance sont représentés sous la forme d'un rapport détaillé et en couleur. Ce rapport permet à l'entrepreneur d'identifier rapidement ses forces et ses faiblesses à l'aide de nombreux graphiques colorés et de plusieurs commentaires textuels. En effet, une codification en différentes couleurs dans les graphiques est utilisée afin de positionner l'entreprise par rapport à son groupe témoin, par exemple le rouge signifiant que l'entreprise est très en retard par rapport à son groupe témoin tandis que le vert signifiant qu'elle est très en avance sur son groupe témoin. De plus, des commentaires textuels sont donnés pour mieux comprendre les résultats du diagnostic contenus dans les graphiques. Finalement, des recommandations sont données à l'entrepreneur afin de l'aider à améliorer la performance de son entreprise. Voici un exemple (figure 2.1) d'une des pages de ce rapport qui illustre les résultats du diagnostic de la performance de l'entreprise du côté de ses ressources humaines.

## RESSOURCES HUMAINES

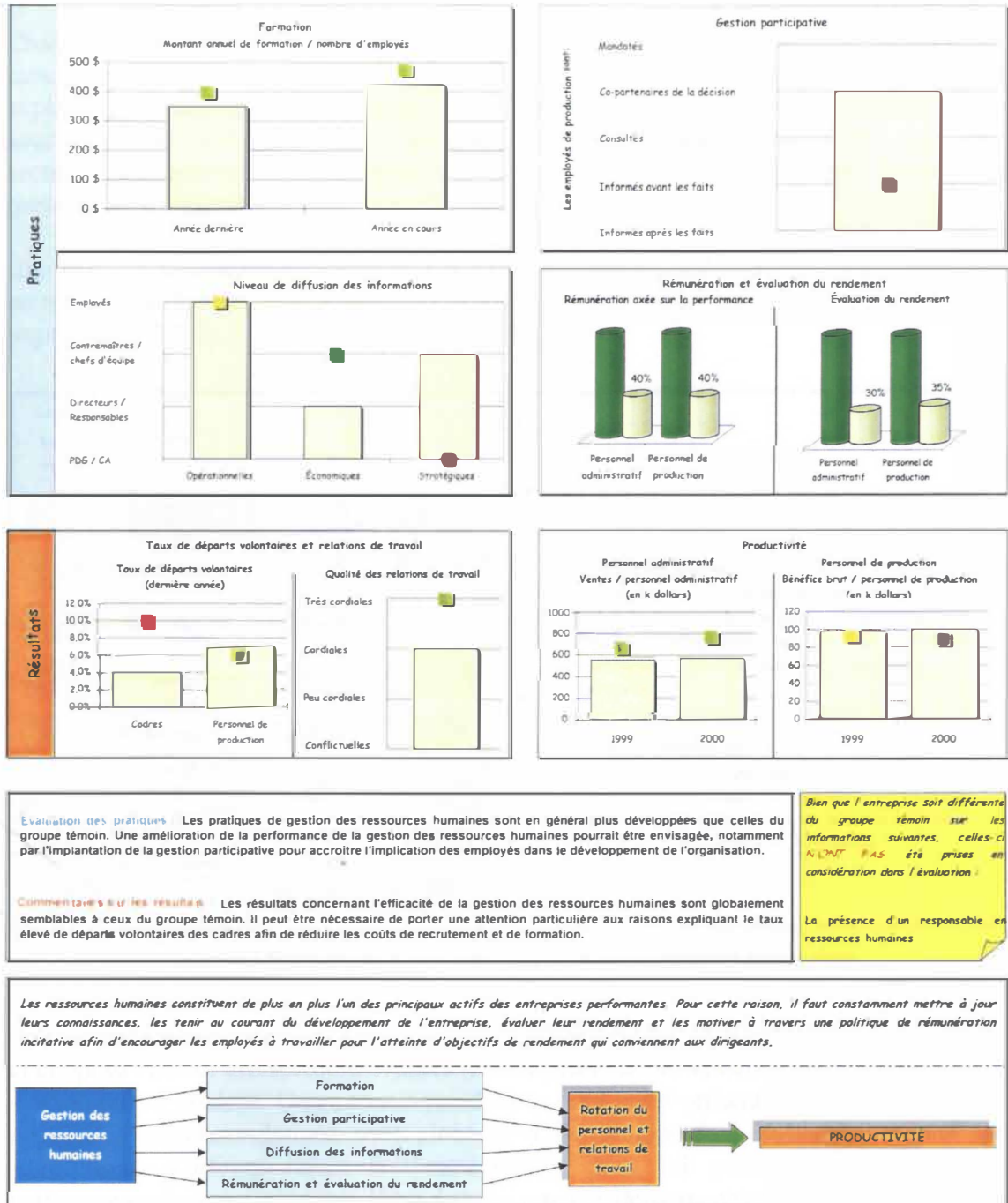
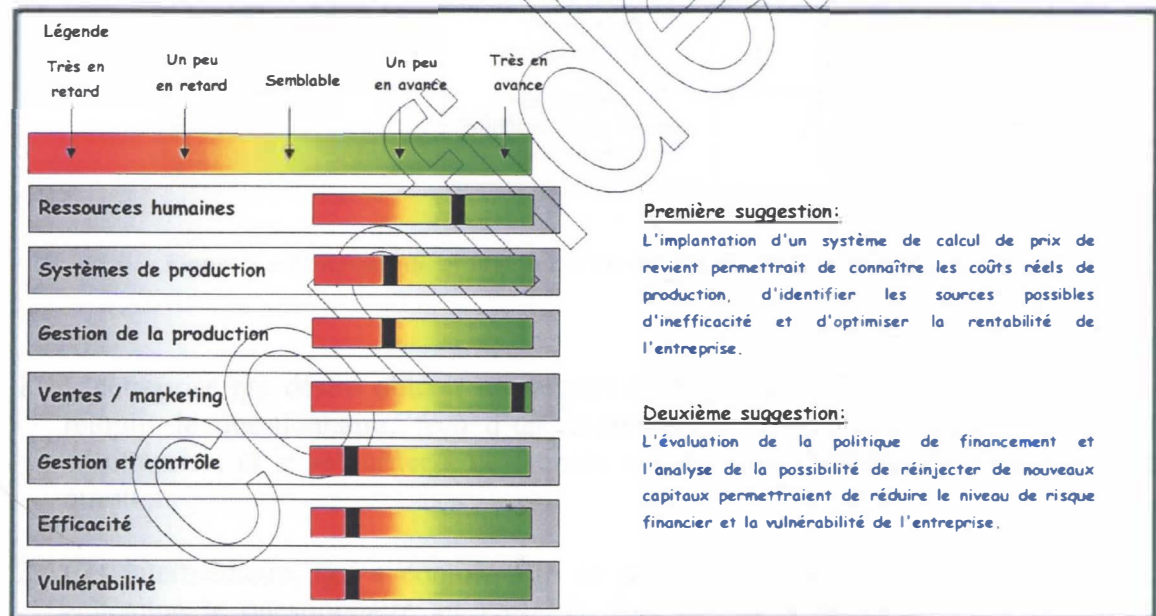


Figure 2.1 Rapport de diagnostic PDG page des ressources humaines

Le rapport complet contient donc les résultats du diagnostic pour chacun des secteurs d'évaluation (les 5 fonctions et 2 sections d'intégration) de la performance d'entreprise énuméré au début de cette section dans le tableau 2.1.

Chacun de ces secteurs possèdent une page de résultats à l'image de la page des ressources humaines vue précédemment dans la figure 2.1, contenant comme il a été expliqué précédemment, graphiques de couleurs, commentaires et recommandations sous forme de texte. Toutefois, les deux derniers thèmes correspondant aux deux sections d'intégration (efficacité et vulnérabilité), possèdent des pages de résultats quelque peu différentes.

Une fois les résultats du diagnostic présentés de manière individuelle pour chacun des secteurs d'évaluation, une évaluation globale du diagnostic est présentée en première page (figure 2.2).

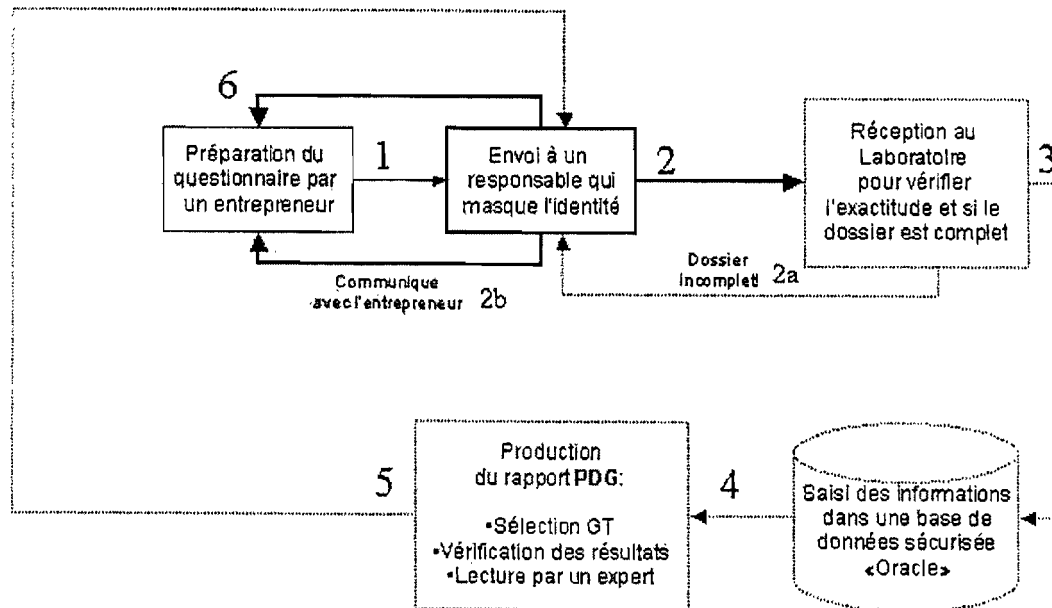


**Figure 2.2 Rapport de diagnostic : performance globale et suggestions**

Cette première page contient le pointage de l'entreprise par rapport à son groupe témoin, et ce, pour chacun des secteurs évalués tel que présenté par le marqueur noir sur la barre synoptique de couleur. Deux recommandations sont aussi présentées sur cette première page. Ces recommandations sont globales à l'ensemble de l'évaluation et sont des suggestions d'actions faites à l'entrepreneur pour lui permettre d'améliorer la performance de son entreprise. Il est à noter que ces deux recommandations ne sont pas générées automatiquement par le système PDG, mais manuellement par l'expert qui analyse les résultats du diagnostic.



Une séquence de plusieurs étapes doit être exécutée afin d'obtenir un rapport de diagnostic PDG. La figure 2.3 illustre ces différentes étapes.



**Figure 2.3 Processus de production d'un diagnostic PDG et du rapport [8]**

1. L'entreprise qui désire obtenir un rapport de diagnostic PDG doit en premier lieu remplir le questionnaire, reçu d'un intermédiaire ou du Groupement des chefs d'entreprise, et le retourner à ce dernier, qui masque l'identité de l'entreprise en question.
2. Cet intermédiaire, qui a comme rôle de protéger la confidentialité du processus, achemine le questionnaire au LaRePE. Des assistants de recherche du LaRePE s'occupent de vérifier la cohérence des réponses et si le questionnaire a bien été rempli. En cas d'erreur dans le questionnaire ou bien s'il n'est pas complet, ce dernier est retourné à l'intermédiaire qui à son tour communique avec l'entreprise pour apporter les corrections nécessaires.
3. Une fois le questionnaire validé, les assistants de recherche font la saisie des données du questionnaire et des états financiers (également reçus), dans une base de données sécurisée Oracle.

4. Le système PDG peut alors être utilisé pour faire le diagnostic de la performance de l'entreprise. L'utilisation de ce système nécessite plusieurs manipulations de la part du personnel du LaRePE.
  - a. Ils doivent tout d'abord faire la création du groupe témoin d'entreprises auquel sera comparée l'entreprise à évaluer. Pour ce faire, un outil maison est utilisé afin d'identifier les différents critères de sélection du groupe témoin. Parmi ces critères, on retrouve les ventes, l'âge, le nombre d'employés, le secteur d'activité, le rythme de croissance, le type de production, les exportations, le ratio d'employés de production et la sous-traitance (voir pages 8-10). Le choix de ces critères peut, soit provenir de l'entreprise à évaluer via le questionnaire, ou bien il peut venir des professionnels du LaRePE, mais ce choix est toujours fait sur mesure pour l'entreprise selon ses besoins et ses orientations stratégiques.
  - b. Une fois ces critères identifiés, le système va chercher dans la base de données toutes les données de toutes les entreprises correspondant aux différents critères de sélection identifiés.
  - c. Le logiciel SAS exécute alors son programme statistique afin d'obtenir toutes les statistiques nécessaires au système PDG pour le diagnostic de l'entreprise (statistique du groupe témoin et de l'entreprise elle-même).
  - d. Le système PDG peut alors être mis en marche afin d'exécuter le diagnostic et la production du rapport.
5. Le rapport de résultats est alors vérifié par les experts du LaRePE afin de s'assurer que les résultats contenus dans ce dernier sont conformes à la réalité et qu'ils ne sont pas erronés. Dans le cas où des manques de précisions seraient trouvés, le rapport est corrigé manuellement. Lorsque le rapport est validé, il est imprimé et envoyé à l'intermédiaire ou au Groupement des chefs d'entreprise, qui est le seul à connaître l'identité réelle de l'entreprise évaluée.
6. Cet intermédiaire expédie finalement le rapport à l'entreprise concernée.

### **2.1.2 Limites du système**

Bien que ce système conçu en 1999 ait été un succès, il n'en demeure pas moins qu'il contient de nombreuses limites nuisant selon le cas, à sa bonne utilisation, à son développement futur, et à sa compréhensibilité. De plus, comme au niveau de sa conception, il n'a pas été construit selon les règles de l'art des systèmes experts et de l'intelligence artificielle. Il est bien important de spécifier que la présentation des limites du système ne constitue en rien une critique sur les choix de conception faits par le passé, mais bien une présentation de ces limites par rapport aux besoins présents et futurs du système.

Voyons en détail les différentes limites du système PDG.

### Décentralisation des connaissances (expertise)

Comme nous le verrons plus loin, un système expert fonctionne à l'aide, entre autres, de faits et de règles. Ces derniers, qui correspondent aux connaissances de l'expert sur le domaine, permettent d'effectuer un diagnostic. Or le système PDG contient bel et bien des connaissances lui permettant d'effectuer le diagnostic de la performance d'une entreprise. Par contre, ces connaissances sont dispersées un peu partout dans l'implémentation sans règles strictes : l'expertise y est donc très mal organisée. Cette mauvaise organisation se traduit par une absence de représentation formelle, centralisée et structurée de tous les éléments de connaissance. Ces derniers se retrouvent un peu partout à travers les cellules des feuilles Excel ainsi qu'à travers les 36 modules du programme VB de diagnostic et de production du rapport PDG comme en témoigne la figure 2.4.

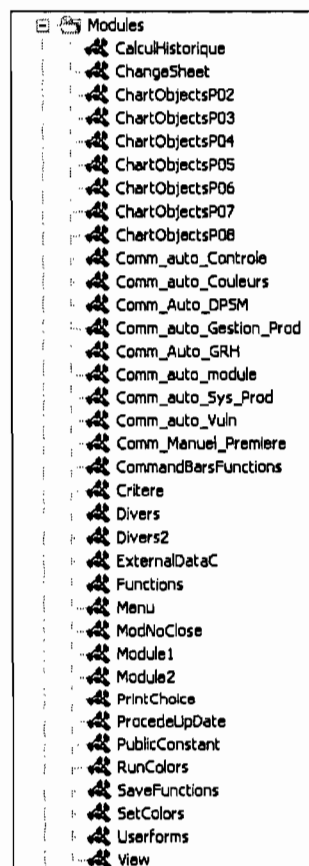


Figure 2.4 Modules du système PDG

Cet état de situation a comme conséquence que la maintenance de cette expertise est très complexe et que le risque d'erreur est grand. Comme c'est le cas pour un expert humain, l'expertise sur un domaine ne cesse de s'améliorer d'année en année. Il en va de même pour un système expert. Par conséquent, les éléments d'expertise d'un système expert se doivent d'être facilement retrouvables et modifiables, ce qui n'est pas le cas pour le système PDG étant donné que ses éléments d'expertise ne sont pas bien centralisés et structurés. Il s'agit donc d'une limite majeure du système PDG. Des éléments d'expertise facilement retrouvables et modifiables seraient un grand atout pour une bonne évolution du système.

#### Utilisation de l'outil Microsoft Excel et manque de flexibilité du système

Dans le dernier point, nous avons vu que les connaissances se retrouvent directement dans l'implémentation du système PDG. La cause de cela est que l'outil utilisé, Microsoft Excel, ne permet pas la création d'une base de connaissances indépendante du reste du système. Ce fort couplage, entre le code du système et les éléments de connaissance, a comme conséquence que le système est « pris dans un pain » et que sa maintenance y est très difficile. De plus, l'implémentation du système a été faite de manière peu orthodoxe. Elle contient énormément de duplication de code VB et la grande utilisation des cellules Excel rend les modifications très périlleuses. En effet, il existe un très fort couplage entre les cellules Excel et le code du système ce qui fait que la modification du contenu ou de l'emplacement d'une seule cellule Excel peut rendre le système inutilisable.

Ce manque de flexibilité, qui est dû à l'outil utilisé et à l'implémentation du système, le rend très peu flexible et par conséquent encore une fois très difficile de maintenance et d'évolution, ce qui constitue une autre limite pour son bon développement futur.

#### Commentaires automatiques du rapport

Les commentaires qui sont donnés sur chaque secteur d'évaluation de l'entreprise sont faits sous forme de phrase complète. Il est très difficile par programmation de faire la génération de phrases bien formulées. La méthode employée fut la concaténation de plusieurs bouts de phrases choisis selon les résultats de l'évaluation de l'entreprise. Cela rend le système encore une fois très peu flexible au niveau de ces phrases. En effet, pour modifier ou rajouter un commentaire, un exercice de programmation doit être effectué, non sans difficultés.

### Caractère non flexible du rapport de diagnostic

La configuration du rapport de diagnostic du système PDG est la même pour toutes les entreprises. Il s'agit encore là d'une limite de flexibilité, car il n'y a aucune possibilité de modifier le rapport pour une entreprise qui le désirerait, aucune possibilité de créer des profils de diagnostics différents. Un profil de diagnostic est une orientation spécifique que prend un diagnostic. Par exemple, une entreprise pourrait vouloir obtenir un profil d'évaluation concernant spécifiquement ses systèmes de production. Or, le système PDG ne permet pas la production d'un diagnostic et d'un rapport d'un tel profil spécifique. Il permet seulement la production du diagnostic et du rapport complet sur tous les secteurs d'activités supportés par ce dernier. De plus, à un niveau plus général, il se peut qu'avec le temps, certaines formes de graphiques soient devenues désuètes et qu'elles demandent une amélioration, voire un changement du graphique. Dans la version actuelle du système PDG, il est pratiquement impossible de faire un changement de graphiques, dû à l'implémentation difficile dont nous avons discuté précédemment. Bref, le caractère fixe et non générique du rapport est une limitation du système.

### Aucune sauvegarde des résultats des diagnostics

Une autre limite du système PDG est qu'aucune sauvegarde des résultats de diagnostic produits n'est faite sous un format de base de données. Cela a pour conséquence qu'il n'existe aucune trace des résultats des diagnostics et qu'il est alors plus difficile de faire évoluer le système et son expertise sur le diagnostic de la performance d'entreprise. Il serait très intéressant de pouvoir conserver les différents résultats amenant à la production de chaque rapport. Comme les résultats du diagnostic, et par conséquent du rapport, peuvent être modifiés lors de sa vérification par un expert, il serait aussi intéressant de pouvoir conserver la trace de tout changement. Cela apporterait des données précieuses à une éventuelle implémentation de technique d'apprentissage automatique et à l'utilisation de techniques de forage de données (« *data mining* »).

### Révision et modification du rapport

Chaque rapport doit être vérifié, validé et s'il y a lieu modifié, par un membre expert de l'équipe du LaRePE. Ce processus se fait à même le rapport de diagnostic produit, ce qui est relativement facile et rapide d'utilisation. Par contre, ce processus de révision et de modification pourrait être directement relié au système avant même la production du rapport, ce qui constituerait une amélioration intéressante.

### Lenteur du système

Une dernière lacune est au niveau de la lenteur du système PDG. En effet, la partie qui nécessite le logiciel SAS pour le calcul des statistiques et du groupe témoin prend environ 2 minutes à s'exécuter. C'est d'autant plus important que cette exécution doit être reprise pour chaque diagnostic. Pour ce qui est du programme même de diagnostic et de production du rapport du système PDG, il prend quant à lui 2 autres minutes à s'exécuter. Ce dernier temps d'exécution est surtout dû à une implémentation non optimale dans le logiciel Microsoft Excel (feuilles Excel et programmation VB).

## **2.2 Objectifs visés**

Pour faire suite aux différentes limites du système PDG qui viennent d'être présentées, voici maintenant la liste des objectifs que visait ce projet de maîtrise.

- Centraliser tous les éléments d'expertise.
- Faciliter les modifications et les améliorations du système.
- Améliorer les commentaires automatiques du rapport de diagnostic.
- Permettre des configurations plus personnalisées du rapport de diagnostic.
- Sauvegarder toutes les informations (données, résultats) nécessaires de chaque diagnostic.
- Faciliter la révision et la modification des rapports de diagnostic.
- Améliorer les temps d'exécution.

Dans ce chapitre, nous avons détaillé la problématique qui entoure le système PDG et fixé les objectifs pour l'élaboration d'une solution afin de pallier aux différentes limites de ce système. Le prochain chapitre consiste en une recension de la littérature qui présentera différents concepts importants de l'intelligence artificielle et qui nous permettra d'élaborer notre solution par après dans ce document.

## CHAPITRE 3

### ÉTAT DE L'ART

## 3.1 Système d'aide à la décision

### 3.1.1 Définition

Dans l'introduction de ce document, le terme DSS a été défini pour désigner un système d'aide à la décision. Comme il a été aussi défini, ce type de système informatique permet à son utilisateur de prendre des décisions plus éclairées et plus sûres sur un domaine en particulier. Le but ultime d'un DSS est d'aider l'utilisateur à prendre une bonne décision et non de prendre la décision à sa place. Voici une définition plus officielle.

« A decision support system (DSS) is a system under the control of one or more decision makers that assists in the activity of decision making by providing an organized set of tools intended to impose structure on portions of the decision-making situation and to improve the ultimate effectiveness of the decision outcome. » [20]

### 3.1.2 Caractéristiques

Voici plusieurs caractéristiques et capacités qu'un bon DSS doit posséder [30] :

- *Utilisé pour des problèmes semi-structurés ou non structurés.* En effet, un problème structuré ne requiert pas l'usage d'un système de ce genre étant donné que le problème est structuré et que par conséquent les actions et les décisions y sont plutôt clairement définies. Il s'agit de décisions répétitives et facilement programmables. Par contre, lors de problèmes semi-structurés et non structurés, les actions et décisions à prendre ne sont pas aussi clairement définies et c'est là qu'entre en jeu l'apport d'un système d'aide à la décision. Par exemple, la gestion d'une demande de crédit de 1000\$ par un client d'une banque est un problème structuré, qui arrive souvent, et dont la décision se fonde toujours sur les mêmes paramètres et les mêmes données. D'un autre côté, la gestion d'une demande de crédit de 1 000 000\$ est un problème qui peut être non structuré, qui n'arrive pas souvent et dont les paramètres de décisions sont loin d'être les mêmes dans tous les cas de ce genre. Un autre exemple de problèmes non

structurés pourrait être l'embauche d'un nouvel employé. C'est une tâche non répétitive et qui demande un raisonnement peu structuré.

- *Peut profiter autant à un seul individu qu'à un groupe d'individus.* C'est-à-dire que selon son domaine d'application et son contexte d'utilisation, le système peut servir autant à un seul individu qu'à un groupe d'individus, pour des décisions qui doivent être prises individuellement ou en équipe.
- *Supporte les 4 phases du processus de prise de décision.* Comme nous l'avons vu dans le chapitre 1, ces phases sont la phase d'intelligence, la phase du design, la phase du choix et la phase de l'implémentation.
- *Supporte une variété de style de prise de décision.* Il existe plusieurs styles de décision : décision directive, décision analytique, décision conceptuelle, décision comportementale. Voici dans la figure 3.1, la description de chacun de ces styles de décision.

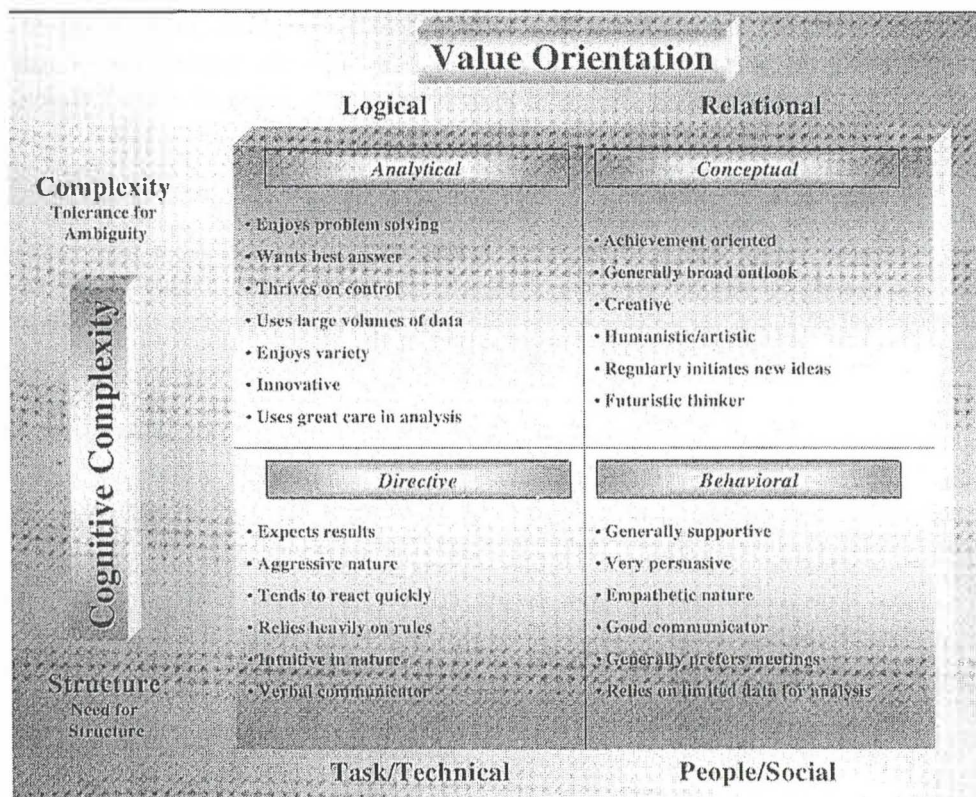


Figure 3.1 Les 4 styles de décision [21]

- *Doit être facilement maintenable et flexible.* C'est-à-dire qu'on peut facilement y ajouter, supprimer, et modifier des éléments. Il s'agit là d'une caractéristique



importante de ce type de système. En effet, les méthodes de raisonnement dans l'analyse de l'aide à la décision peuvent venir qu'à changer avec le temps, tout comme un humain peut changer ses méthodes d'analyses, et tout ça, dans le but d'améliorer ses prises de décisions. Par conséquent, le système se doit d'être assez flexible afin qu'on puisse facilement le modifier (ajout, suppression, et modifications d'éléments).

- *Doit être convivial pour l'utilisateur.* L'utilisation du système ne doit pas être compliquée. Par conséquent, l'interface d'utilisation doit être la plus simple possible afin que l'utilisateur puisse configurer facilement et rapidement les différents paramètres du système afin que ce dernier puisse conseiller et aider le preneur de décision. De plus, l'aide à la décision (conseils, recommandations) doit se faire de la manière la plus simple possible. Par exemple, l'utilisation de graphiques peut grandement aider cet élément.
- *Doit avoir comme but d'améliorer l'efficacité concernant la précision, la rapidité et la qualité de la décision au lieu de l'efficacité concernant le coût de la décision.* Bref, le système doit rendre des propositions de décisions efficaces, mais doit laisser de côté le coût de la décision et à la place prendre en considération la précision, la rapidité et la qualité de la décision.
- *Le DSS est là pour guider le preneur de décision et non pour le remplacer.* Il s'agit là de l'élément fondamental de ce type de système. Un DSS accompagne le preneur de décision, lui propose des éléments et des pistes de décisions, mais ne prend pas la décision à sa place. Ces propositions et pistes de décisions peuvent par exemple se retrouver dans un rapport produit par le système.

### 3.1.3 Composition

Tel qu'illustré dans la figure 3.2, un DSS est composé de plusieurs sous-systèmes.

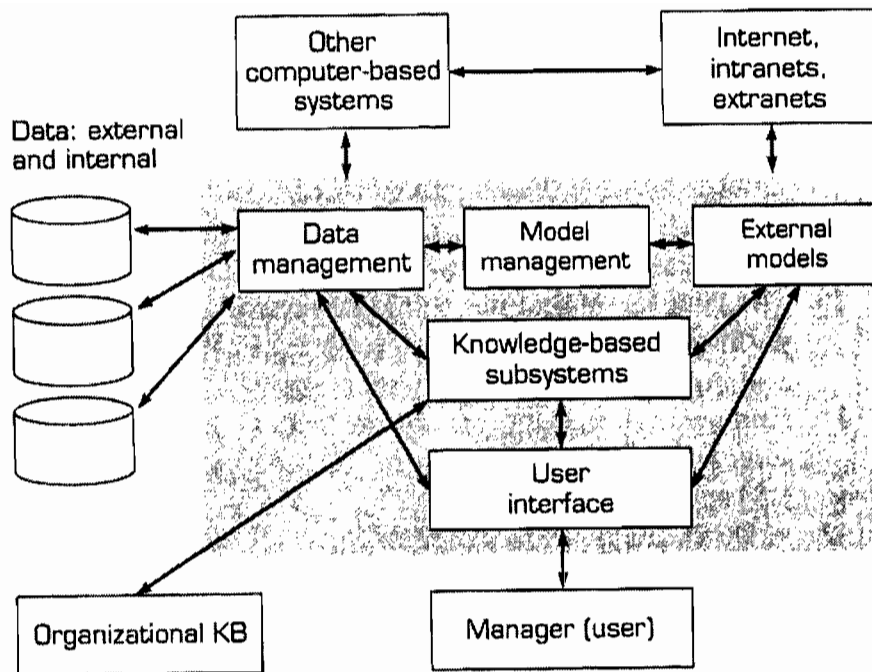


Figure 3.2 Architecture habituelle d'un DSS [30]

Voici une description sommaire de ces différents sous-systèmes :

**Sous-système de gestion des données** (« *Data management subsystem* »). Ce sous-système est composé des éléments suivants :

- Une base de données dans laquelle on retrouve les données en rapport avec le domaine et le contexte du DSS.
- Un système de gestion de la base de données. Il s'agit du système qui créera, accèdera et mettra à jour la base de données du DSS.
- Un catalogue de variables. Il s'agit d'un catalogue ou dictionnaire contenant plusieurs informations sur les données de la base de données.
- Un langage permettant de faire des requêtes pour accéder, manipuler, et interroger les données.

**Sous-système de gestion des modèles** (« *Model management subsystem* »). Ce sous-système gère tous les modèles qui peuvent servir entre autres, à la capacité d'analyse du DSS. Un modèle dans un DSS est la modélisation la plus fidèle possible du problème qui doit être résolu et ses solutions envisageables.

**Sous-système de gestion des connaissances** (« *Knowledge-based management subsystem* »). Ce sous-système gère toutes les connaissances (expertise) que le DSS

peut utiliser. C'est avec ces connaissances que le DSS pourra exécuter son analyse afin d'accompagner le preneur de décision.

***Sous-système de l'interface utilisateur (« User interface subsystem »).*** C'est à l'aide de ce sous-système que l'utilisateur communique avec le DSS. L'efficacité d'un DSS se mesure entre autres, par le degré de facilité d'utilisation qu'offre l'interface utilisateur du DSS.

### **3.1.4 Domaines et exemples d'application**

On retrouve les DSS dans plusieurs domaines. La gestion d'entreprises est un domaine dans lequel l'utilisation d'un DSS est bien souvent primordiale. Ce type de système est essentiel pour une entreprise qui opère dans un environnement compétitif. Plus une entreprise prendra des décisions rapides et sûres, plus elle sera efficace vis-à-vis de ses concurrents. Pour une entreprise n'utilisant pas de DSS, les décisions prises rapidement ne seront peut-être pas les meilleures et d'un autre côté, les décisions réfléchies trop longuement peuvent être bonnes, mais seront peut-être en retard vis-à-vis des concurrents. De plus, les décisions appliquées du type « essai-erreur » ne sont plus souhaitables puisqu'une petite erreur dans une décision peut avoir des répercussions sur plusieurs autres niveaux d'une entreprise. Ce type de décision « essai-erreur » peut donc être remplacé dans plusieurs cas par un DSS qui, de manière très rapide, simulerait les résultats de plusieurs scénarios en lien avec la décision à prendre et donnerait les résultats au preneur de décision pour lui démontrer la décision qui donnerait les meilleurs résultats. C'est donc dire que la quantité d'effort et le temps mis pour prendre une bonne décision se trouvent à être réduits et par conséquent la productivité de l'entreprise se voit augmenter. Un autre grand avantage de l'utilisation d'un DSS est la quantité de connaissances que l'on peut y stocker. Contrairement à un humain, un tel type de système peut conserver en mémoire un nombre incroyable de connaissances en y altérant aucunement la rapidité avec laquelle il peut faire le traitement de ces connaissances. Bref, l'utilisation d'un DSS permet de prendre des décisions efficaces, c'est-à-dire précises, rapides et de qualité.

Ces différentes raisons pour lesquelles la présence d'un DSS est très avantageuse dans le domaine de la gestion d'entreprises sont pour la plupart les mêmes pour tous les autres domaines dans lesquels un DSS peut être important. Voici des domaines dans lesquels l'utilisation d'un DSS peut être d'une grande utilité [20].

- ***Environnement.*** Un DSS peut rendre un bon coup de main dans l'évaluation d'impacts de certains problèmes sur l'environnement. Par exemple, des DSS ont été développés depuis la fin des années 1980 dans le but de supporter la prise de décision sur les réserves d'eau, et ce, autant au niveau commercial que gouvernemental.

- *Agriculture.* Des systèmes d'aide à la décision pour la bonne gestion de systèmes d'irrigations aidant à l'amélioration de production de nourriture dans des régions en manque d'eau naturelle est un exemple parmi plusieurs dans ce domaine. Les DSS, à un niveau plus large, aident grandement à l'amélioration globale des processus de production de l'agriculture. Par exemple, les processus pour assister l'identification de maladies et de traitements dans les récoltes.
- *Aviation.* Le domaine de l'aviation utilise les systèmes d'aide à la décision pour de multiples raisons. Par exemple, les compagnies aériennes peuvent en faire utilisation pour l'organisation de l'horaire des vols. L'aviation militaire utilise aussi cette technologie. C'est le cas pour le système d'aide à la décision PADAL de la U.S Navy [28]. Ce système a été développé afin d'aider l'atterrissage des avions de chasse sur un porte-avion. Pour chaque atterrissage d'avion, un officier du porte-avion guide le pilote par contact radio dans son approche finale. Par contre, ce travail n'est pas simple, pour plusieurs raisons dont celle que le porte-avion peut tanguer de 10 pieds vers le haut et de 10 pieds vers le bas. Le système PADAL aide donc l'officier à prévoir la trajectoire de l'avion et à prendre les meilleures décisions, qui doivent bien souvent être prises rapidement.
- *Sylviculture.* Ce type de système peut être d'une grande aide dans les décisions concernant la bonne évolution des forêts (reboisement, financement de la gestion des forêts, etc.).
- *Industrie manufacturière.* Les DSS peuvent être d'une grande aide dans ce domaine. Plusieurs types de décisions en industrie manufacturière peuvent être aidés d'un DSS (approvisionnement du matériel, financière, investissement, etc.).
- *Médecine.* Des systèmes comme MYCIN et DENDRAL [12] servent dans la médecine depuis plusieurs années, autant pour le diagnostic de maladies et de traitements (ces systèmes sont plus associés à des systèmes experts comme nous le verrons plus loin) que pour la gestion du personnel, des patients, du matériel médical, etc. Plusieurs DSS et systèmes experts, classés par domaine de médecine, se retrouvent sur le site Web suivant : <http://www.computer.privateweb.at/judith/>
- *Support organisationnel.* Comme nous l'avons vu précédemment dans cette section, les DSS peuvent aider grandement dans le support organisationnel. La rapidité et la qualité de plusieurs types de décision peuvent ainsi être améliorées (ressources humaines, systèmes de production, ventes et marketing, etc.).
- *Industrie alimentaire.* Un DSS peut être d'une grande utilité dans le domaine de l'industrie alimentaire. C'est le cas par exemple du système FoodPro®

(« <http://www.foodpro.com/> »). Ce système d'aide à la décision est d'une grande utilité pour les entreprises œuvrant dans le service alimentaire en offrant un grand nombre de fonctionnalités. Parmi ces dernières, on retrouve l'aide pour la gestion de l'inventaire des stocks, pour l'achat des stocks, pour la planification de menus, pour l'analyse nutritionnelle des menus, et pour l'analyse financière de l'entreprise.

## 3.2 Système expert

### 3.2.1 Définition

Le terme système expert a été défini comme étant un système informatique qui utilise des connaissances humaines saisies dans un ordinateur afin de résoudre des problèmes normalement résolus par des experts. Un système expert tend donc vers le remplacement du raisonnement humain, à la différence qu'il essaie d'être plus rapide et plus efficace que le raisonnement humain qui est bien souvent trop lent devant des situations trop complexes. L'élément central d'un système expert est la base de connaissances, dans laquelle se retrouve toute l'expertise qu'un expert humain possède sur le domaine d'application du système expert. Cette base de connaissances est jumelée à un moteur d'inférence (ou de raisonnement) qui permet au système expert de raisonner sur son domaine et ainsi tendre vers le raisonnement de l'expert humain (voir figure 3.3).

Voici deux définitions plus officielles d'un système expert.

« A computer-based application that employs a set of rules based upon human knowledge to solve problems that requires human expertise. » [20]

« A computer program whose performance is guided by specific, expert knowledge in solving problems. » [27]

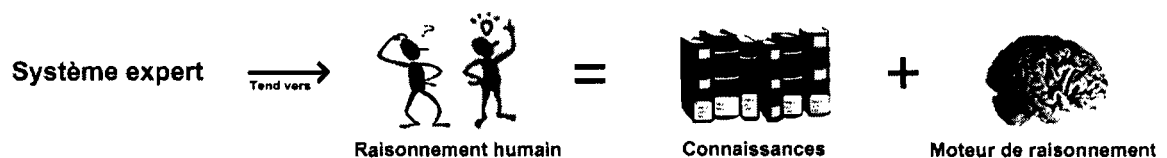


Figure 3.3 Vulgarisation en schéma d'un système expert

### 3.2.2 Caractéristiques

Les systèmes experts sont utiles dans un grand nombre de situations [20].

- Besoin de faire le diagnostic de problèmes dans un contexte donné.
- Besoin de comprendre la nature d'une situation en particulier.
- Besoin de prédire les répercussions d'un événement courant ou futur.
- Besoin de contrôler un processus ou une activité particulière.
- Besoin de recommandations ou d'une solution pour un problème particulier.

Bien entendu, un système expert n'est pas infallible. Comme n'importe quel expert humain, il peut faire des erreurs. Un certain nombre des erreurs provenant de l'expert humain est dû à la fatigue d'analyse, la trop grande charge de travail, le manque de temps, etc. Avec l'utilisation d'un système expert, ce genre d'erreur est éliminé, ce qui diminue le nombre d'erreurs possibles. D'un autre côté, une limitation d'un système expert par rapport à un expert humain est qu'il n'utilise pas le sens commun, le « bon sens », l'intuition, contrairement à un expert humain. Cependant, les bénéfices de l'utilisation de ce type de système sont nombreux [20].

- Augmente la rapidité dans la prise de décision.
- Augmente la productivité des experts humains.
- Améliore la consistance dans les décisions.
- Améliore la compréhension des décisions à l'aide d'explication.
- Améliore la gestion de l'incertitude dans les décisions.
- Formalisation de la connaissance sur le domaine du système expert.

### 3.2.3 Composition

Voici maintenant l'architecture commune d'un système expert.

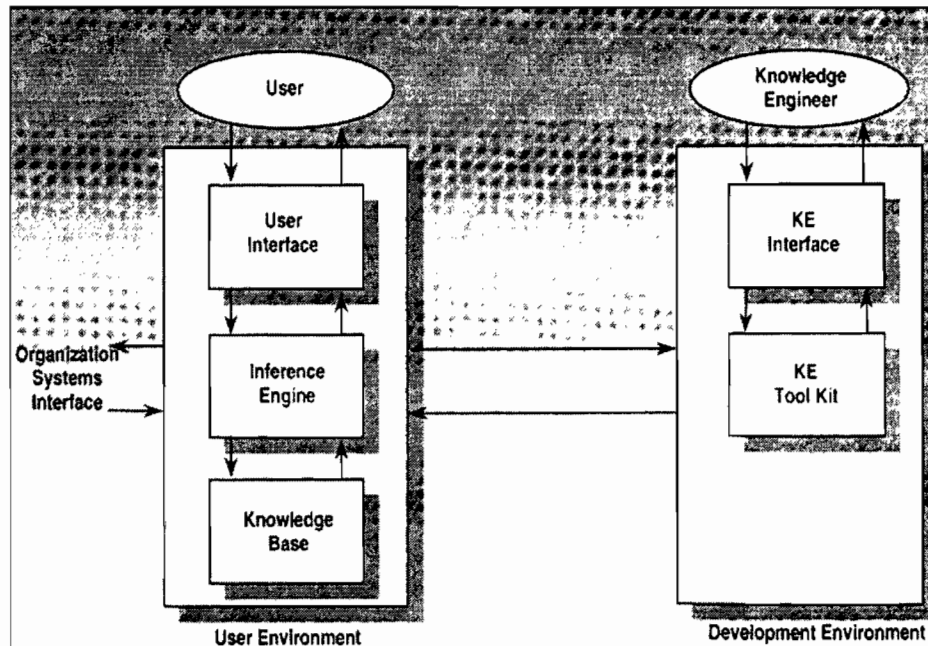


Figure 3.4 Architecture habituelle d'un système expert [20]

Tel qu'illustré dans la figure 3.4, voici les composantes principales d'un système expert :

- *La base de connaissances.* Comme nous le verrons en détail dans la section 3.3, la base de connaissances sert à l'organisation et la centralisation de tous les éléments de connaissance du système expert. Il s'agit de la base du système et sa qualité influera grandement sur les capacités du système expert.
- *L'engin d'inférence.* Cette partie du système expert donne la capacité de raisonnement de ce dernier. L'engin d'inférence utilise les connaissances contenues dans la base de connaissances afin d'appliquer le raisonnement propre au système expert. Il est le cerveau du système.
- *Programme utilisateur.* Ce programme permet à l'utilisateur d'interagir avec le système expert. Tout dépendant du type et du contexte du système expert, le programme utilisateur offre différentes possibilités comme par exemple des possibilités de configuration.

Finalement, la même figure 3.4 nous montre les deux principaux utilisateurs du système expert : l'utilisateur même des fonctionnalités du système expert (via le programme utilisateur) et l'ingénieur des connaissances. Ce dernier étant celui qui, à travers divers outils de travail et interfaces, fait la maintenance du système et de sa base de connaissances.

### 3.2.4 Domaines et exemples d'application

Tout comme les DSS (voir section 3.1.4), les systèmes experts sont utilisés dans un très grand nombre de domaines. Ultimement, tous les domaines où une expertise humaine existe peuvent faire l'objet d'un système expert. Voici une liste de systèmes experts connus [12, 20, 23]:

- DENDRAL (1965-1983). Détermine des structures chimiques de molécules non identifiées.
- MYCIN (1972-1980). Diagnostic et recommandation de traitements pour des infections bactériennes du sang.
- XCON (1980 à aujourd'hui). Permet de déterminer la configuration appropriée de systèmes informatiques par rapport au besoin d'un consommateur.

Ces systèmes, datant de plusieurs années, sont des exemples de systèmes populaires qui ont jeté les bases dans la construction de systèmes experts. Bref, les systèmes experts de nos jours se comptent par milliers et sont présents dans tous les domaines. En voici quelques exemples :

- COLOSSUS®. Système expert utilisé par plusieurs compagnies d'assurances dans le monde afin d'aider l'assureur dans des réclamations pour dommages corporels. (<http://www.csc.com/industries/insurance/mds/mds221/408.shtml>)
- HEPAXPERT III / WWW. Système expert médical permettant l'analyse et l'interprétation de sérologie d'hépatite A et d'hépatite B. (<http://medexpert.imc.akh-wien.ac.at/hepax/>)
- PEPID (« *Portable Emergency Physician Information Database* »). Ce système expert a été conçu pour aider les médecins à diagnostiquer rapidement les problèmes médicaux et de drogue arrivant en urgence à l'hôpital. Faisant suite aux diagnostics, des recommandations des premiers traitements médicaux nécessaires à donner sont proposés par le système. Comme il est utilisé en situation d'urgence, le système expert PEPID est utilisé le plus souvent sur des petits ordinateurs de poche (« *palm* »). (<http://www.pepid.com/>)
- Plusieurs systèmes experts et DSS, classés par domaine de médecine, se retrouve sur le site Web suivant : <http://www.computer.privatweb.at/judith/>
- Des exemples de systèmes experts reliés au domaine de l'agriculture se retrouve sur le site Web « *The central lab for agricultural expert systems* ». (<http://www.claes.sci.eg/Home/Welcome.html>)



### **3.2.5 Système d'aide à la décision versus système expert**

Un système d'aide à la décision (DSS) et un système expert sont donc, selon leur définition, deux entités différentes visant des buts différents. Le premier vise à aider et accompagner l'humain dans ses décisions, et l'autre vise à vouloir remplacer l'humain dans ses décisions. Par contre, il existe des systèmes hybrides utilisant la technologie à la fois des DSS et des systèmes experts. Le cas le plus fréquent du système hybride est le DSS qui possède une base de connaissances complète permettant le raisonnement d'un système expert.

## **3.3 Base de connaissances**

### **3.3.1 Définition**

Une base de connaissances contient toutes les connaissances spécifiques sur un domaine et constitue un élément essentiel pour tout système expert. En effet, pour prendre une bonne décision et faire de bons raisonnements, il faut se baser sur des connaissances de bonne qualité et bien organisées. Il va donc de pair que les connaissances d'un système expert soient organisées dans une base de connaissances. Des connaissances bien organisées peuvent être définies comme étant facilement utilisables, retrouvables et modifiables dans la base de connaissances. De plus, mieux organisées sont les connaissances, moins une modification d'une d'entre elles a des répercussions sur les autres. Voici deux définitions plus officielles :

« A collection of facts, rules, and procedures organized into schemas. The assembly of all the information and knowledge of a specific field of interest. » [22]

« The knowledge base is the repository for the knowledge used by an expert system. The rules and hints for guiding the search for solutions. » [27]

### **3.3.2 Niveaux de connaissances**

Il existe deux niveaux de connaissances [30] :

- Connaissances de surfaces
- Connaissances profondes

Les *connaissances de surfaces* correspondent aux connaissances traitant d'un sujet très spécifique. Par exemple, une connaissance disant que dans le cas où le réservoir d'essence d'une auto est vide, cette auto ne démarrera pas. Ce niveau de connaissance limite grandement la capacité du système à donner des explications à l'utilisateur, et de plus, il apparaît bien souvent insuffisant pour des problèmes plus complexes.

Les *connaissances profondes* correspondent aux connaissances très détaillées sur un sujet. Cela inclut les émotions, le « bon sens », l'intuition, etc. Ce niveau de connaissances est très difficile à informatiser, voire jusqu'à maintenant impossible. Cependant, il est possible de choisir un niveau de connaissances plus profond que pour les connaissances de surfaces, un niveau intermédiaire entre les deux.

### 3.3.3 Catégories de connaissances

On peut aussi classer les connaissances en 3 grandes catégories [30] :

- Connaissances déclaratives
- Connaissances procédurales
- Méta connaissances

Les *connaissances déclaratives* servent à définir des faits, ce que sont les choses. Cette catégorie de connaissances est considérée comme étant du niveau des connaissances de surfaces. Ce sont habituellement les premières connaissances à acquérir dans le processus d'acquisition de connaissances. Ce processus sera vu un peu plus loin dans ce chapitre à la section 3.3.5.

Exemple : Le salaire d'une personne X est de 85 000\$ par année.

Les *connaissances procédurales* servent à représenter de quelle manière les éléments interagissent dans plusieurs circonstances. Les règles de décision sont le type de connaissances que l'on retrouve le plus souvent dans cette catégorie. Bref, ce type de connaissances sert à savoir de quelle manière utiliser les connaissances déclaratives, comment y faire de l'inférence.

Exemple : Calculer le salaire d'une personne. Si son salaire est supérieur à 100 000\$, alors accepter automatiquement tout prêt d'une valeur de 10 000\$ et moins. Si son salaire est entre 60 000\$ et 100 000\$, calculer d'abord le niveau d'endettement, etc.

Les *méta connaissances* sont des connaissances sur les connaissances. Elles servent à venir raffiner les connaissances générales sur le domaine d'expertise.

Exemple : On peut avoir la connaissance qu'une personne a une cote de crédit de A. Une méta connaissance pourrait venir raffiner cette connaissance en donnant plus de connaissances sur ce qu'est une cote de crédit et comment elle se calcule.

### 3.3.4 Types de connaissances

Comme le montre la figure 3.5, les connaissances utilisées en intelligence artificielle peuvent être de plusieurs types.

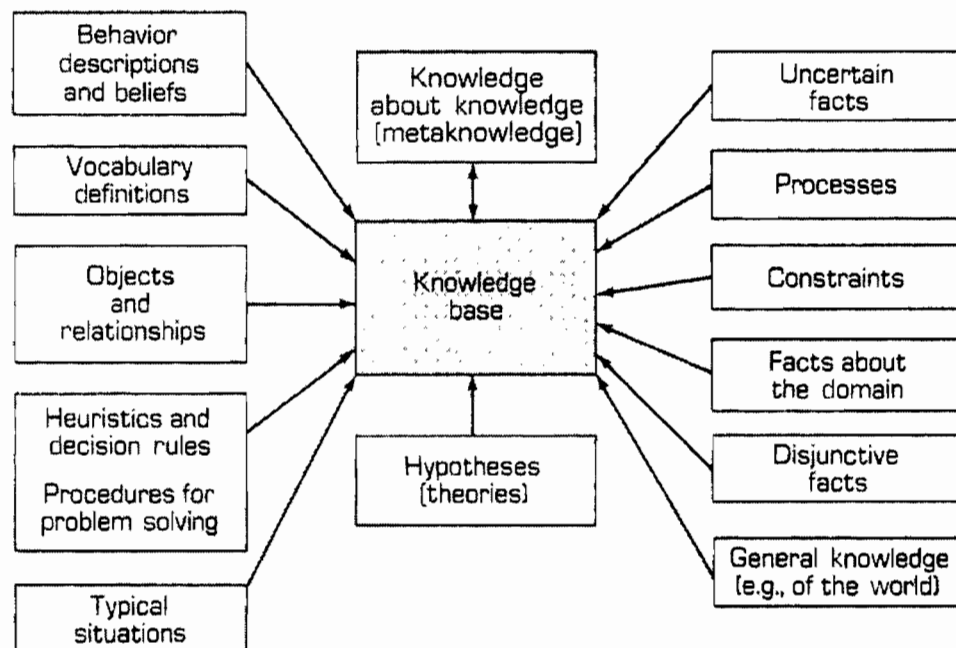


Figure 3.5 Différents types de connaissances [30]

Bien qu'il existe un grand nombre de types de connaissances, tous ne sont pas toujours les plus utilisés. Voyons un peu plus en détail les types de connaissances les plus souvent utilisés.

- *Faits*. Les faits sont des faits vérifiés qui sont utilisés pour l'inférence (le raisonnement).

Exemple : L'âge moyen des stocks de l'entreprise X est de 38 jours.

- *Règles de décision.* Les règles de décision sont des règles qui utilisent les différentes connaissances, dont les faits. Ces règles seront utilisées par le moteur d'inférence (que nous verrons plus loin dans ce document à la section 3.3.5.4) dans le but d'appliquer le raisonnement.

Exemple : Si le ratio entreprise X / groupe témoin concernant l'âge moyen des stocks est entre 0 et 0.5, alors on peut affirmer que l'entreprise X est grandement inférieure à son groupe témoin.

- *Procédures.* Les procédures sont aussi utilisées dans le but d'appliquer le raisonnement, mais elle ressemble à de la programmation conventionnelle. On utilise entre autres les procédures lorsque des calculs complexes sont nécessaires au raisonnement.
- *Méta connaissances.* Les méta connaissances sont, comme nous l'avons vu précédemment, des connaissances qui viennent raffiner d'autres connaissances afin d'élargir les connaissances générales sur le domaine d'expertise.

Exemple : Prenons l'exemple du fait qui utilise la notion d'âge moyen des stocks. Nous pourrions vouloir créer une méta connaissance sur cette notion d'âge moyen des stocks afin de la définir (par exemple, définir la manière de le calculer).

### 3.3.5 Ingénierie de la connaissance

Maintenant que nous avons défini la notion de base de connaissances et son contenu, étudions de quelle manière une base de connaissances de qualité est construite.

Le processus de construction d'une base de connaissances se nomme « knowledge engineering (KE) » ou bien ingénierie de la connaissance. Comme on peut le voir sur la figure 3.6, ce processus inclut 5 grandes activités majeures [30].

1. Acquisition des connaissances
2. Validation des connaissances
3. Représentation et saisie des connaissances
4. Inférence
5. Explication et justification

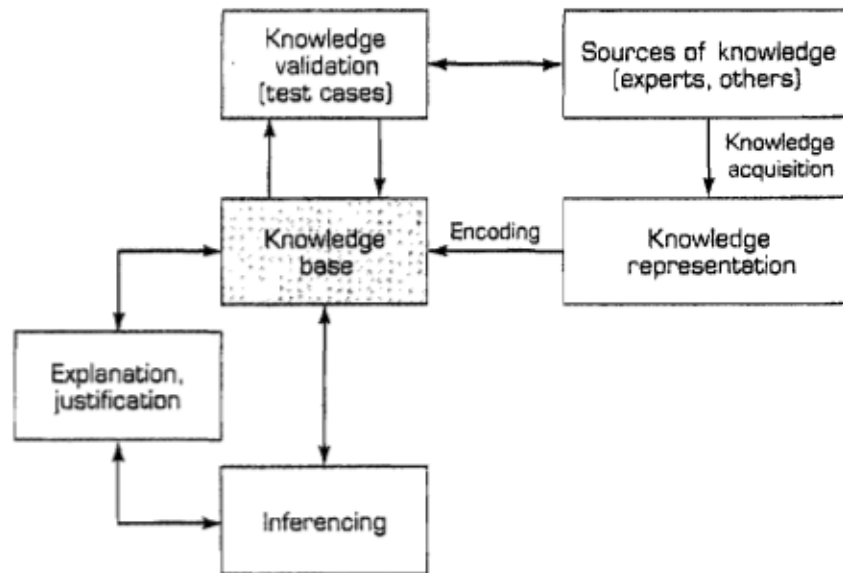


Figure 3.6 Processus d'Ingénierie de la connaissance [30]

### 3.3.5.1 Acquisition des connaissances

Cette première activité consiste à acquérir les connaissances qui seront utiles dans la base de connaissances. Plusieurs chercheurs s'entendent pour dire que l'acquisition des connaissances est une activité critique dans l'élaboration d'une bonne base de connaissances, et par conséquent, d'un bon système expert. Tout comme l'expert humain qui a besoin de bonnes connaissances pour bien raisonner, un système expert en a lui aussi besoin.

Précédemment, nous avons vu que les connaissances pouvaient être de plusieurs types : faits, règles de décision, procédures, etc. Mais où trouve-t-on ces connaissances ?

Les connaissances peuvent provenir de plusieurs sources d'expertise telles que d'experts humains, de livres, de documents, de capteurs, de fichiers électroniques, d'images, etc. Bien souvent, on retrouve beaucoup de ces sources dans des bases de données ou bien sur Internet. De plus, on peut diviser les sources de connaissances en deux types, les sources documentées et non documentées. Les *sources documentées* proviennent de documents, tandis que les *sources non documentées* font référence à la connaissance humaine qui peut ne pas être bien documentée.

L'acquisition des connaissances n'est pas une tâche facile. Il peut être très difficile pour un expert de bien exprimer ses connaissances. Bien souvent, ses connaissances proviennent de son expérience dans le domaine et il peut être très difficile de l'exprimer

« sur papier », spécialement lorsque cette expérience provient de sensations, de la pensée, de « *feelings* ».

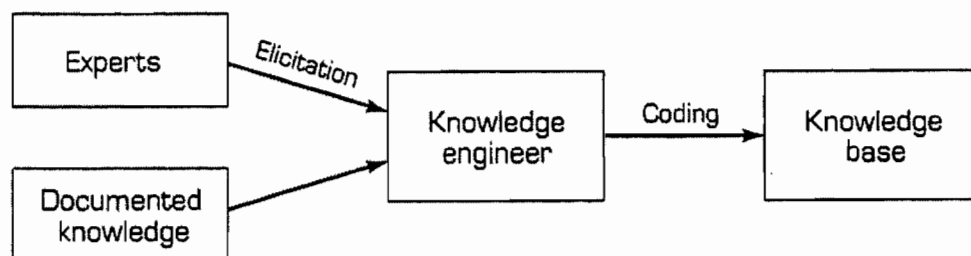
Dans un monde idéal, on peut compter 4 participants à l'acquisition des connaissances :

- L'expert
- L'ingénieur de connaissances
- Le constructeur du système
- Le futur utilisateur

Ces participants ne voient pas de la même manière les connaissances, ne communiquent pas avec la même terminologie, et de plus ils n'ont pas les mêmes habiletés. C'est pourquoi il peut parfois être difficile de s'entendre sur l'acquisition des connaissances et sur le futur système. Dans un monde idéal, l'expert devrait prendre un rôle actif dans le processus d'acquisition des connaissances, tandis que l'ingénieur des connaissances devrait agir plus à titre d'enseignant sur la manière de bien structurer les connaissances et à titre d'intermédiaire entre l'expert et le futur utilisateur.

Il existe plusieurs types de méthodes pour faire l'acquisition de connaissances :

- *Méthodes manuelles.* Dans ce type de méthodes, l'ingénieur de connaissances devra aller chercher les connaissances manuellement à partir d'entrevues d'expert ou bien d'autres sources pour ensuite aller les coder à la main dans la base de connaissances. Ce type de méthodes n'est pas très rapide, coûte cher et est bien souvent imprécis. Voir figure 3.7.



**Figure 3.7 Méthode manuelle d'acquisition des connaissances [30]**

- *Méthodes semi-automatiques.* Ce type de méthodes se situe entre les méthodes manuelles et automatiques, et ce, en minimisant soit le besoin de l'expert, soit le besoin de l'ingénieur de connaissances.

- *Méthodes automatiques.* Ce type de méthodes tend à minimiser les rôles de l'expert et de l'ingénieur de connaissances. Le terme « automatique » doit être pris avec du recul. Il est pratiquement impossible, pour l'instant, d'automatiser complètement le processus d'acquisition des connaissances. Il y aura toujours un humain d'impliqué dans ce processus, mais pas nécessairement un expert et un ingénieur de connaissances.

Comme les méthodes manuelles et semi-automatiques dépendent beaucoup de la qualité de l'ingénieur de connaissances et de l'expert, les méthodes automatiques sont à privilégier. Plusieurs méthodes automatiques existent :

- « *Machine learning* ». Il s'agit de méthodes qui permettent à un système d'acquérir et de structurer sous forme de règles des connaissances directement à partir d'un document, d'une base de données, etc. Plusieurs outils commerciaux existent pour faire ce type de traitement automatique.
- *Induction automatique de règles.* Il s'agit du processus par lequel des règles sont générées par un programme informatique à partir de cas exemples dont on connaît le bon résultat final. Il existe aussi plusieurs outils commerciaux pour ce type de traitement.
- *L'induction interactive.* Dans ce cas, un programme aidera l'expert, de manière interactive, à faire de l'acquisition et de l'induction de connaissances.
- « *Case-based reasoning* ». Il s'agit de compiler tous les cas historiques déjà résolus dans la base de connaissances et de s'en servir pour résoudre de futurs cas similaires.
- *Agents intelligents.* L'utilisation d'agents intelligents [23] peut permettre d'acquérir des connaissances très rapidement à moindre coût.

### **3.3.5.2 Validation des connaissances**

C'est dans cette activité que les connaissances acquises sont validées et vérifiées. Pour ce faire, il y aura utilisation de tests, dont les résultats seront présentés aux experts pour en vérifier la précision. Cette activité se terminera lorsque la qualité des connaissances sera acceptable.

La *validation* permet de savoir si la base de connaissances est la bonne par rapport à son contenu.

La *vérification*, quant à elle, permet de savoir si la base de connaissances a été construite d'une bonne manière.

Plusieurs critères de validation sont présentés dans la figure 3.8.

<i>Measure or Criterion</i>	<i>Description</i>
Accuracy	How well the system reflects reality, how correct the knowledge is in the knowledge base
Adaptability	Possibilities for future development, changes
Adequacy (or completeness)	Portion of the necessary knowledge included in the knowledge base
Appeal	How well the knowledge base matches intuition and stimulates thought and practicability
Breadth	How well the domain is covered
Depth	Degree of detailed knowledge
Face validity	Credibility of knowledge
Generality	Capability of a knowledge base to be used with a broad range of similar problems
Precision	Capability of the system to replicate particular system parameters, consistency of advice, coverage of variables in knowledge base
Realism	Accounting for relevant variables and relations, similarity to reality
Reliability	Fraction of the ES predictions that are empirically correct
Robustness	Sensitivity of conclusions to model structure
Sensitivity	Impact of changes in the knowledge base on quality of outputs
Technical and operational validity	Quality of the assumed assumptions, context, constraints, and conditions, and their impact on other measures
Turing test	Ability of a human evaluator to identify whether a given conclusion is made by an ES or by a human expert
Usefulness	How adequate the knowledge is (in terms of parameters and relationships) for solving correctly
Validity	Knowledge base's capability of producing empirically correct predictions
<i>Source: Adapted from B. Marcot, "Testing Your Knowledge Base." AI Expert. Aug. 1987.</i>	

Figure 3.8 Critères de validation des connaissances [30]

### 3.3.5.3 Représentation et saisie des connaissances

Dans cette activité, les connaissances acquises et validées seront organisées et codées dans la base de connaissances. Il est très important que les connaissances soient bien structurées et organisées dans la base de connaissances. Comme nous l'avons mentionné



précédemment, pour obtenir des connaissances bien structurées et organisées, ces dernières se doivent d'être facilement utilisables, retrouvables, et modifiables dans la base de connaissances. Cela permettra aussi aux activités d'inférence et de raisonnement d'être plus rapides et de meilleure qualité.

Tous les types de représentations des connaissances partagent deux caractéristiques :

- Elles peuvent être programmées à l'aide d'un langage de programmation courant et elles peuvent être stockées en mémoire.
- Leur modélisation permet d'utiliser les connaissances afin de faire du raisonnement.

Voici quelques types de représentations de connaissances :

- *Les règles de production (« Production rules »)*. Les connaissances y sont représentées sous la forme de paires condition-action.

Forme → SI une condition survient ALORS une action doit être déclenchée.

Chaque règle de production implémente une partie de l'expertise de la base de connaissances. Il est bien important de comprendre qu'il existe deux types de règles. Il y a les règles de connaissances (règles déclaratives) et les règles d'inférences (règles procédurales). Les règles de connaissances représentent les faits sur le problème. Les règles d'inférences quant à elles, représentent les règles qui serviront à régler le problème, ce sont les règles de raisonnement.

Il y a plusieurs avantages à utiliser les règles de productions :

- Les règles sont facilement compréhensibles.
- L'inférence y est facilitée.
- Les modifications et la maintenance sont relativement faciles.
- Bien souvent, les règles sont indépendantes les unes des autres.

Il y a aussi des limitations à utiliser les règles de productions :

- Lorsque l'on est en présence de connaissances plus complexes qui requièrent un très grand nombre de règles, l'utilisation et la maintenance du système peuvent y être plus difficiles.
- Comme elles sont faciles d'utilisation, les concepteurs de base de connaissances ne considèrent pas assez les autres types de représentations dans leur choix de représentation.

- « *Frames* ». La représentation « **frames** » est une application de la programmation orientée objet. Il s'agit de représenter chaque objet dans la base de connaissances par un « **frame** ». Les connaissances dans un « **frame** » sont divisées en « **slots** ». Chaque « **slot** » peut contenir des connaissances déclaratives (faits) ou bien des connaissances procédurales. On peut donc dire que les « **slots** » sont les attributs qui décrivent l'objet qui est représenté par le « **frame** ». De plus, chaque « **slot** » peut contenir une ou plusieurs **facettes**. Les facettes servent à décrire les connaissances sur les attributs (« **slots** »). De plus, comme cette représentation est directement influencée de l'orienté objet, il est possible de créer une hiérarchie de « **frames** ».

Voici dans la figure 3.9, un exemple de hiérarchie de « **frames** » décrivant une base de connaissances sur des véhicules:

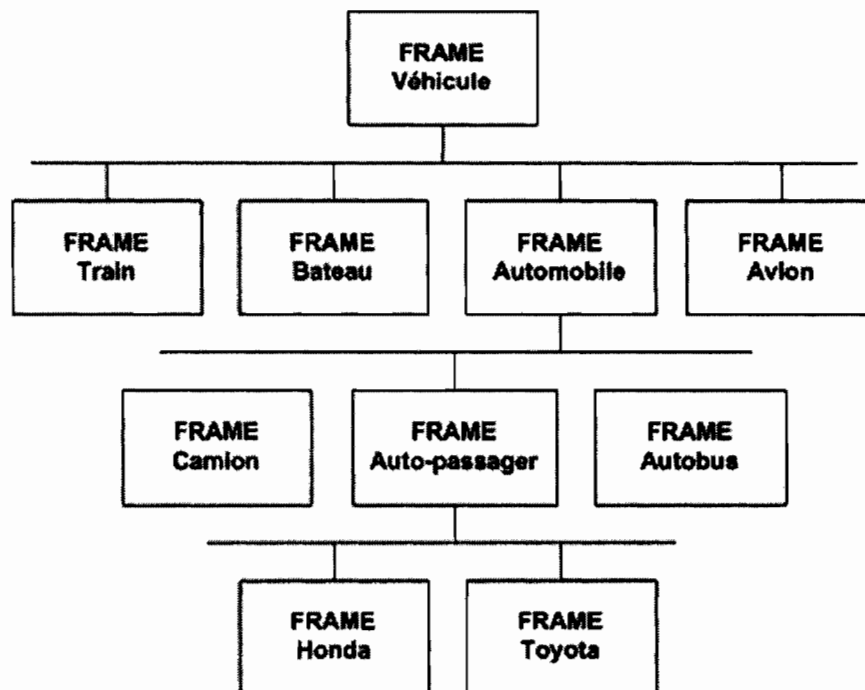


Figure 3.9 Exemple de hiérarchie de « **frames** » représentant des véhicules [30]

#### 3.3.5.4 Inférence

L'activité d'inférence implique l'utilisation d'un logiciel afin de permettre au système de faire de l'inférence (du raisonnement) sur les connaissances et sur le problème qu'il tente de résoudre. Cette activité peut se faire lorsque la base de connaissances est bien organisée.

Le logiciel utilisé pour faire de l'inférence est appelé « engin d'inférence » ou « moteur d'inférence ». L'engin doit connaître des règles d'inférences pour savoir de quelle manière diriger sa recherche dans la base de connaissances. Dans ce cas, l'engin d'inférence parcourt la base de connaissances en décidant quelles règles il doit analyser, quelles alternatives il doit éliminer et lesquelles correspondent bien au problème. Bref, le moteur d'inférence utilise toutes les connaissances de la base de connaissances afin d'appliquer le raisonnement. C'est le cerveau du système expert!

Il existe plusieurs techniques en intelligence artificielle pour faire de l'inférence. En voici quelques-unes :

- « *Forward and backward chaining* ». Il s'agit des deux méthodes qui permettent de faire de l'inférence dans un système basé sur des règles (« rule-based system »). Dans le cas du chaînage avant (« forward chaining »), il s'agit de débiter avec une information de base sur la situation (un fait) et d'essayer de parcourir les différentes règles de la base de connaissances afin de trouver une conclusion probable. Il s'agit de règles de la forme cause  $\rightarrow$  conséquence (if-then). D'un autre côté, dans le chaînage arrière (« backward chaining »), il s'agit de débiter avec ce qui est ou ce qui va arriver dans la situation (la conclusion), et d'essayer de parcourir les différentes règles de la base de connaissances afin de trouver une cause probable. Cette dernière méthode sert souvent pour des cas de diagnostic.

Par exemple, nous avons les règles de décisions suivantes :

Règle #1 :  $A \rightarrow B$

Règle #2 :  $B \rightarrow C$

Si nous connaissons le fait de départ A, nous pouvons à l'aide de ces règles par chaînage avant, trouver la conclusion C.

Si nous connaissons la conclusion C, nous pouvons à l'aide de ces règles par chaînage arrière, trouver la cause A.

- *Arbre d'inférence*. Un arbre d'inférence se compare grandement à un arbre de décision. Chaque règle est composée d'une condition et d'une conclusion, et est représentée par des nœuds dans l'arbre d'inférence. Des branches permettent de connecter les différentes règles. De plus, on y retrouve des opérateurs AND et OR pour bien structurer les règles. L'utilisation de ce type d'arbre permet de bien visualiser le processus de raisonnement d'un problème. Voici dans la figure 3.10 un petit schéma simple d'un arbre d'inférence.

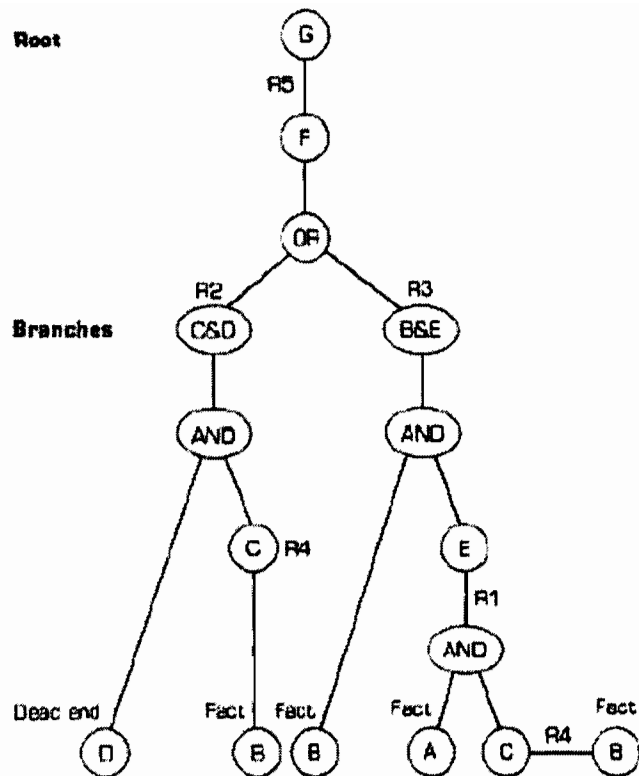


Figure 3.10 Exemple d'un arbre d'inférence [30]

- *Inférence avec les « frames »*. Il s'agit d'une méthode de raisonnement un peu plus compliquée que le raisonnement à l'aide de règles.

Pour faciliter l'inférence, les « frames » peuvent être classifiés selon leur application :

- En « frame » de situations pour représenter les connaissances de ce à quoi on s'attend dans certaines situations.
- En « frame » d'actions pour représenter les actions qui doivent être exécutées sous des conditions spécifiques (procédure, « triggers », etc.).
- En « frame » de causes pour représenter des combinaisons de « frames » de situations et d'actions décrivant des relations cause à effet.

Il y a deux façons habituelles d'implémenter l'inférence avec les « frames ». Utilisation de règles similaires aux règles de production et utilisation du

raisonnement hiérarchique. Bien souvent, il y a combinaison de ces deux manières. Voir [30] pour plus de détails.

- « *Model-based reasoning* ». Ce type de technique est très utile dans le diagnostic de problèmes d'équipements complexes. Dans ce type de systèmes, on retrouve un modèle détaillé de ce qui doit être diagnostiqué. Ce modèle est ensuite utilisé afin de trouver la cause des problèmes (diagnostic). Ce type de méthode est souvent combiné avec d'autres méthodes d'inférences.
- « *Case-based reasoning* ». Il s'agit de trouver des solutions pour de nouveaux problèmes à l'aide de solutions qui ont déjà servi pour d'autres problèmes dans le passé. Voici le processus global de ce type de raisonnement :
  - L'engin d'inférence trouve en mémoire des cas de problèmes similaires au problème courant.
  - Il adapte ensuite la ou les anciennes solutions trouvées pour avoir une bonne correspondance avec le problème courant.

Ce type de méthode est très efficace dans les problèmes d'une grande complexité.

### **3.3.5.5 Explication et justification**

Il s'agit d'inclure à la base de connaissances une capacité d'explication et de justification sur l'inférence faite et par conséquent sur les solutions proposées par le système. Par exemple, le système pourrait être capable de répondre au « pourquoi » qu'une information précise est donnée par le système ou bien au « comment » dont le système est arrivé à une certaine conclusion. Ajouter une capacité d'explication à un système comporte beaucoup d'avantages. En voici quelques-uns :

- Rend le système plus intelligent.
- Permet de découvrir des imperfections dans les règles et autres connaissances de la base de connaissances.
- Explique des situations que l'utilisateur ne prévoit pas.
- Réconforte et assure l'utilisateur sur le bien-fondé des recommandations fournies par le système.

La plupart des outils de développement qui aident à construire des composantes d'explications utilisent les deux explications suivantes :

- *Pourquoi?* (« why? »). Cette question est posée par l'utilisateur après que le système ait demandé une information à ce dernier. Par exemple, supposons que pour effectuer son raisonnement, le système demande le revenu annuel de l'utilisateur. Avant de répondre, l'utilisateur peut vouloir demander au système dans quel but il a besoin de cette information. C'est à cela que sert la question « pourquoi ? ».
- *Comment?* (« how? »). Cette question est posée par l'utilisateur lorsqu'il veut connaître de quelles manières une certaine conclusion a été trouvée par le système. Généralement, afin d'expliquer une certaine conclusion, le système écrit les différentes règles qu'il a utilisées pour y arriver. Dans un système un peu plus complexe et plus intelligent, les règles utilisées pour arriver à la conclusion peuvent être écrites dans une explication utilisant du langage naturel.

Il est possible de générer des explications de deux manières :

- De manière *statique*. Il s'agit d'associer à chaque question une réponse statique. Il suffira juste à imprimer ce texte lorsque la question associée sera posée. Ce n'est évidemment pas très efficace, car ça implique de connaître les réponses à l'avance à toutes les questions. Ce qui peut s'avérer une tâche très lourde pour un grand système. De plus, la maintenance peut y être très difficile et finalement ce n'est pas très « intelligent » comme principe.
- De manière *dynamique*. Il s'agit de reconstruire les explications selon la manière dont les règles ont été utilisées. C'est un principe un peu plus complexe à mettre en application, mais ça rend le système beaucoup plus « intelligent » et beaucoup plus facile à maintenir. Le niveau de dynamisme des explications est relié au niveau de profondeur et de détail des règles.

Dans ce chapitre, nous avons fait le tour des principaux concepts d'intelligence artificielle entourant le développement de DSS, de systèmes experts, de base de connaissances et de l'ingénierie des connaissances, etc. Ces concepts seront d'une grande utilité dans la compréhension de la solution qui a été élaborée et qui est présentée dans le prochain chapitre de ce document.

## CHAPITRE 4

# RÉINGÉNIERIE DES CONNAISSANCES ET DÉVELOPPEMENT D'UN NOUVEAU SYSTÈME

### 4.1 Présentation globale du système

Le système PDG, un système de diagnostic de la performance d'entreprise réalisé par le LaRePE de l'UQTR en 1999, a été présenté dans le deuxième chapitre de ce document. Nous y avons vu que malgré sa grande utilité, ce système contient plusieurs faiblesses importantes au niveau de l'implémentation, nuisant à sa flexibilité et à ses possibilités d'améliorations dans le futur. Les faiblesses du système PDG et sa conception, qui est loin des règles de l'art de l'intelligence artificielle, constituaient donc la problématique de mon travail de maîtrise. De quelle manière obtenir un système expert PDG construit selon les règles de l'art de l'intelligence artificielle, sans les limites actuelles, lui permettant ainsi de grandes possibilités pour le futur et permettant au LaRePE de continuer à développer son expertise sur domaine de la performance d'entreprise?

Pour répondre à ces objectifs, il a été pensé de conserver le système PDG et de le modifier afin de l'améliorer et d'en corriger les faiblesses. Par contre, vu sa conception et la teneur de ces faiblesses (voir le chapitre 2 pour les détails), il aurait été beaucoup trop complexe de travailler à même ce système PDG et pratiquement impossible de lui greffer de futures composantes d'intelligence artificielle.

La solution devait donc passer par la conception d'un tout nouveau système. Ce dernier devait être construit selon les règles de l'art de l'intelligence artificielle, tout en tenant compte des principes du système PDG. Voici une présentation générale de ce nouveau système que nous avons nommé le **système PDGII**.

Le système PDGII, tout comme son prédécesseur, est un système permettant le diagnostic de la performance d'entreprise, à l'exception qu'il utilise des techniques adéquates d'intelligence artificielle ce qui en fait un vrai système expert. Voici dans les prochains paragraphes de cette section, une présentation globale des éléments qui caractérisent ce nouveau système expert.

#### Une base de connaissances

Une base de connaissances a été construite afin d'y regrouper toute l'expertise qui était dispersée un peu partout dans le système PDG. Cette expertise correspond à toutes les

connaissances dont les experts humains ont besoin pour le diagnostic de la performance d'entreprise. Un long processus d'ingénierie de la connaissance fut nécessaire afin de concevoir cette base de connaissances, qui constitue l'élément central de ce nouveau système PDGII.

### Un moteur d'inférence

Un moteur d'inférence est utilisé afin d'appliquer le raisonnement sur la base de connaissances, ce qui permet le diagnostic.

### Un entrepôt de données

De plus, le nouveau système utilise un entrepôt de données pour les différents calculs statistiques, calculs qui étaient jadis faits par le logiciel SAS dans le système PDG. Ce nouvel entrepôt de données, qui a été conçu par Mathieu Dugré, un diplômé de la maîtrise en informatique faisant partie de l'équipe du LaRePE et maintenant professionnel de recherche au même laboratoire, est venu grandement améliorer le traitement et la manipulation des nombreuses données qui sont utilisées au LaRePE [7]. Cet entrepôt de données a permis l'abandon de l'utilisation du logiciel SAS, procure plus de flexibilité, et réduit les délais, ce qui constitue de grandes améliorations pour le nouveau système. Cet entrepôt de données a aussi permis la sauvegarde de toutes les informations (résultats, données, expertise utilisée, etc.) issues des diagnostics du système PDGII, ce qui constitue une autre amélioration intéressante.

### Une interface utilisateur conviviale

Une interface utilisateur a aussi été produite pour le système PDGII. Cette interface, très conviviale et accessible à l'aide d'un navigateur Web, est venue améliorer la facilité de production d'un diagnostic d'une entreprise et de son rapport.

Grâce à ces éléments du nouveau système, les problèmes de flexibilité, autant au niveau du système, de l'expertise, que du rapport, ont été réglés. La prochaine section de ce chapitre présente en détail ces différents éléments de solutions du système PDGII et nous verrons donc de quelles manières ce nouveau système a gagné en flexibilité et comment il a corrigé les limites de son prédécesseur.



## 4.2 Éléments de solutions du nouveau système

Dans cette section, chaque élément de solution sera présenté en détail. Pour chaque élément, la solution même sera présentée ainsi que tous les aspects de son développement et de sa technique. Voici tout d'abord dans la figure 4.1, une vue globale de l'architecture du nouveau système PDGII.

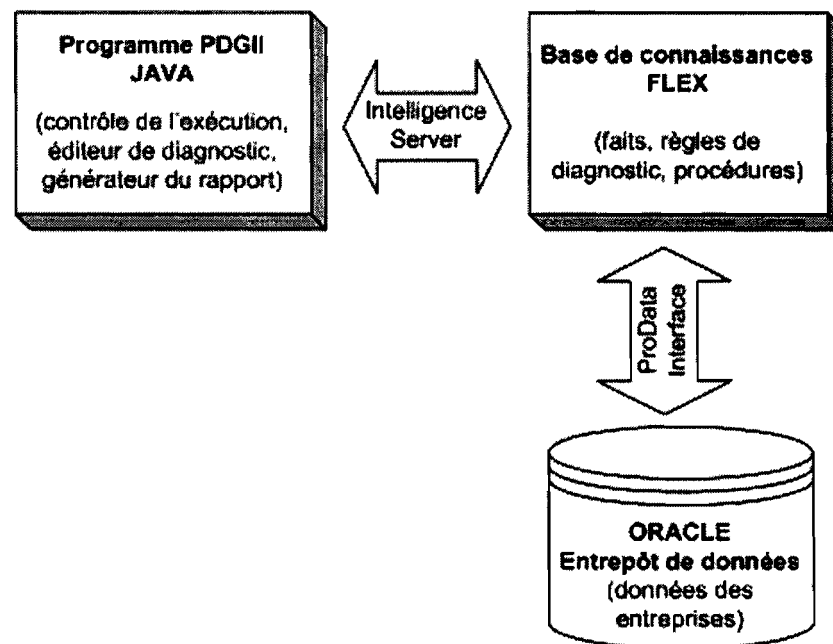


Figure 4.1 Architecture du système PDGII

### 4.2.1 La base de connaissances

L'élément de solution le plus important du nouveau système est sans contredit la base de connaissances. En comparaison avec le système PDG, la centralisation et la structuration de toutes les connaissances dans une même base de connaissances est une amélioration majeure pour le nouveau système de diagnostic de la performance d'entreprise. Il s'agit d'une amélioration majeure pour des raisons de flexibilité, d'améliorations futures et d'utilisation de techniques d'intelligence artificielle, comme nous le verrons en détail dans ce chapitre. La venue de cette base de connaissances était donc une nécessité pour le développement d'un système expert suivant les règles de l'art de l'intelligence artificielle.

#### **4.2.1.1 Conception et technologie**

La conception de la base de connaissances a nécessité un long travail d'analyse et d'ingénierie de la connaissance dans lequel toutes les étapes prescrites à la section 3.3.5 ont été suivies.

La première étape, celle de l'**acquisition des connaissances**, fut une étape cruciale dans la conception de cette base de connaissances. Comme la majeure partie des connaissances se trouvait déjà dans le système PDG, la grande partie du travail d'acquisition des connaissances a consisté à l'étude en profondeur de ce système, à la recherche de tous les éléments de connaissance. Les connaissances du système PDG étaient éparpillées et non-structurées dans le code de programmation VB et dans les cellules Excel du système. En effet, une analyse rigoureuse, des 36 modules du programme VB du système PDG ainsi que des nombreuses feuilles de calculs Excel et leurs cellules, a été effectuée. La grande difficulté était de n'oublier aucun élément de connaissance.

##### Critères d'évaluation

Lors de ce travail, on pouvait déjà entrevoir une possibilité de structuration de la connaissance. En effet, une grande partie des connaissances acquises était les critères d'évaluation (ou indicateurs de performance) du diagnostic PDG. Un critère d'évaluation est un critère permettant d'évaluer la performance d'une entreprise sur un élément bien précis. Voici quelques exemples de critères d'évaluation acquis lors du travail d'acquisition des connaissances :

- Âge moyen des stocks
- Efficacité de la production
- Qualité des relations de travail
- Productivité du personnel administratif
- Investissement dans la production
- Degré de maîtrise technologique

Il est à noter qu'un critère d'évaluation peut être autant quantitatif que qualitatif. Un diagnostic complet contient donc une multitude de critères d'évaluation.

##### Commentaires de diagnostic

L'autre partie importante des connaissances acquises dans le système PDG était celle des commentaires de diagnostic. Tous les commentaires et recommandations textuels,

ainsi que leurs règles d'utilisations constituent des éléments d'expertise très importants pour la base de connaissances du futur système PDGII.

Bref, les connaissances acquises sont constituées d'une part de tous les critères d'évaluation permettant de faire un diagnostic complet de la performance d'une entreprise. Ces critères d'évaluation sont composés quant à eux de divers éléments d'expertises telles des pondérations, des règles de calculs, etc. D'une autre part, on retrouvait tous les commentaires et recommandations textuels, ainsi que leurs règles d'utilisation.

Nous verrons plus en détail un peu plus loin la représentation et la structure qu'ont prise ces diverses connaissances acquises.

Une fois les connaissances acquises du système PDG ainsi que des experts du LaRePE, elles ont dû être validées par ces mêmes experts lors l'étape de **validation des connaissances**, afin de s'assurer qu'elles étaient encore bonnes, valides et complètes.

L'étape suivante de conception de la base de connaissances fut celle de **représentation et de saisie des connaissances** dans laquelle il fallait représenter les connaissances acquises et validées sous une forme structurée et pouvant être saisie dans la base.

### Type de représentation

Afin de bien représenter les connaissances dans la base, il fallait décider du type de représentation désirée qui s'adapterait le mieux aux connaissances acquises. Or, nous avons vite trouvé des similitudes entre ces dernières et le type de représentation en « frame ». En effet, tout comme la définition de ce type de représentation qui veut que l'on fasse des groupes d'expertise avec les éléments de connaissance, il était facile de grouper les différents éléments de connaissance acquis. Voici deux groupes qui émergeaient :

- Un groupe d'expertise contenant les différentes connaissances des critères d'évaluation. De plus, ces différents critères pouvaient faire partie de sous-groupes encore plus précis comme par exemple celui des ressources humaines ou encore celui des systèmes de production.
- Un autre groupe d'expertise contenant les différentes connaissances des commentaires de recommandation du diagnostic.

Bien entendu, c'était loin des groupes finaux, mais ça nous donnait un bon aperçu du bien-fondé de l'utilisation du type de représentation en « frame » que nous avons choisi sans hésitation.

### À la recherche d'un outil pour la représentation en « frame »

Il fallait ensuite trouver un logiciel permettant la construction et la gestion de la base de connaissances. Ce logiciel devait bien sûr permettre le type de représentation en « frame ». Comme plusieurs logiciels existent déjà pour accomplir cette tâche, il n'a pas été nécessaire d'en construire un. Par contre, une étude approfondie de logiciels existants a dû être faite afin de trouver le meilleur logiciel pour les besoins du nouveau système PDGII.

Après une vaste recherche, deux logiciels ont fait l'objet d'une étude plus approfondie : Protégé et Flex.

Le **logiciel Protégé** [26] est un logiciel qui permet de faire l'édition de bases de connaissances et d'ontologies. Il a été conçu par un groupe de recherche de l'Université Stanford aux États-Unis. Protégé permet, tel que souhaité, de construire des bases de connaissances sous forme de « frames ». Le grand avantage est que Protégé offre une interface utilisateur très conviviale afin d'éditer une base de connaissances. De plus, il s'agit d'un logiciel gratuit.

Par contre, un désavantage de ce logiciel est que la sauvegarde de la base de connaissances dans une base de données, fonctionnalité intéressante, se fait dans une seule table, et de la manière dont les données se retrouvent dans cette table, il n'est pas facile d'en faire une utilisation future, par exemple pour faire du forage de données (« *data mining* ») sur la base de connaissances. De plus, Protégé possède un désavantage important du côté de son inférence. Protégé ne possède pas de moteur d'inférence qui lui est propre. Pour pouvoir faire du raisonnement avec ce logiciel, il faut se tourner vers des logiciels « plugs-in ». Après étude de plusieurs de ces logiciels de moteur d'inférence, celui qui semblait le mieux adapté à Protégé était Algernon [11]. Algernon permet de faire de l'inférence sur une base de connaissances Protégé à base de « frames ». Un désavantage de ce moteur d'inférence est le langage dans lequel doivent être écrites les règles d'inférence. Il s'agit d'un langage du style « fonctionnel » qui est difficilement compréhensible pour un non-informaticien. Par exemple, une simple addition « 3 + 4 » s'écrit « (+ 3 4) » en langage fonctionnel.

Voici un exemple d'une règle écrite en langage style « fonctionnel » ressemblant au langage Algernon :

```
(defrule regle01)
(refrigerator light on) (refrigerator door open)
=>
(assert (refrigerator food spoiled)))
```

La compréhension, l'ajout et la modification de règles dans ce style de langage peuvent devenir assez difficiles dans un système complexe.

Même s'il s'agit d'un logiciel très convivial et très intéressant à première vue, dus à ces nombreux désavantages, nous n'avons pas choisi le logiciel Protégé. Nous nous sommes plutôt tournés vers le logiciel Flex, pour de multiples raisons qui seront décrites dans les prochains paragraphes.

Le **logiciel Flex** [16] est un outil de la suite de logiciels Win-Prolog de la compagnie Logic Programming Associates Ltd [15] (LPA) et permet de faire la création de systèmes experts complets. Il est donc possible de faire la création d'une base de connaissances sous forme de « frame » avec Flex et de faire le raisonnement sur cette base à l'aide du moteur d'inférence de Flex. Afin de mieux comprendre son choix pour le nouveau système PDGII, voici une présentation du logiciel Flex et des fonctionnalités qu'il offre.

#### Édition de base de connaissances avec Flex

Bien entendu, Flex permet l'édition de base de connaissances. Le type de représentation qu'il utilise est la représentation sous forme de « frames » (voir section 3.2.5.3). L'édition d'une base de connaissances se fait dans un fichier texte ASCII standard (.ksl) à même l'environnement de développement de l'outil Flex à l'aide du langage propre à Flex : le langage Knowledge Specification Language (KSL). Il s'agit d'un langage basé sur le langage Prolog. Cependant, le langage KSL de Flex est plus facile de compréhension et d'utilisation puisque ce dernier se rapproche beaucoup du langage naturel de l'anglais.

Voici un exemple de déclaration d'un frame représentant un étudiant à l'aide du langage utilisé par Flex :

```
frame etudiant ;  
    default nationalite is canadien and  
    default domaine is informatique and  
    default no_cycle is 1 and  
    default education is moyenne .
```

Voici maintenant un exemple de déclaration d'une instance de ce frame etudiant:

```
instance maria is a etudiant ;  
    nationalite is espagnol and  
    domaine is ingenierie and  
    cycle is 1 .
```

### Règles d'inférence avec Flex

Flex a la capacité de faire du raisonnement sur ses bases de connaissances directement à l'aide de son propre moteur d'inférence. Pour ce faire, il suffit d'éditer, dans le même fichier texte que dans lequel on retrouve la base de connaissances, les différentes règles dont tiendra compte le moteur d'inférence. Ce dernier permet l'utilisation du chaînage avant et du chaînage arrière. Pour le chaînage avant, il faut utiliser le mot-clé **rule** tandis que pour le chaînage arrière, il faut utiliser le mot-clé **relation**.

Voici un exemple d'une règle avec chaînage avant selon la syntaxe de Flex :

```
rule verification_education_etudiant
  if S is an instance of etudiant and S`s cycle > 1
  then S`s education becomes elevee .
```

Cette règle `verification_education_etudiant` vérifie le cycle des étudiants. Si un étudiant a un cycle plus grand que 1, son éducation se verra mise à la valeur « `elevee` ».

Il est aussi possible de faire l'utilisation du raisonnement par chaînage arrière. La grande utilité que peut avoir l'utilisation du chaînage arrière avec Flex, est qu'il est possible, grâce à la syntaxe de Flex, d'utiliser des paramètres dans la relation. Ce qui n'est pas possible dans les règles de chaînage avant. Voici un exemple de chaînage arrière avec Flex :

```
relation education_elevee( S )
  if S is an instance of etudiant and S`s no_cycle > 1
```

Cette relation `education_elevee` prend en paramètre une instance `s` et vérifie si le but d'avoir une éducation élevée est respecté.

### Justification de règles avec Flex

Il est possible de rajouter une justification de la règle avec le mot clé **because**. Par exemple :

```
rule verification_education_etudiant
  if S is an instance of etudiant and S`s no_cycle > 1
  then S`s education becomes elevee
  because L'éducation est élevée car un faible pourcentage
  d'individu dans la population ont des études de cycles
  supérieurs ;
```

### Pondération des règles avec Flex

Dans l'optique où plusieurs règles peuvent être satisfaites, il est possible d'attribuer un score à chaque règle avec le mot-clé **score** afin qu'un choix puisse être fait quant à savoir quelle règle doit être exécutée.

```
rule verification_education_etudiant
  if S is an instance of etudiant and S`s no_cycle > 1
  then S`s education becomes elevee
  because L'éducation est élevée car un faible pourcentage
  d'individu dans la population ont des études de cycles supérieurs
  score 2.
```

### Traitement de l'incertitude avec Flex

Il est aussi possible de faire du raisonnement avec traitement de l'incertitude. Cela peut être fait grâce à l'outil Flint [17] qui fonctionne très bien avec Flex.

Flint supporte 3 types de traitement de l'incertitude :

- La logique floue (« Fuzzy logic »)
- La mise à jour Bayésienne (« Bayesian updating »)
- Les facteurs de certitudes (« Certainty factors »)

Nous verrons plus en détail l'outil Flint et la notion de traitement de l'incertitude dans le chapitre 6 de ce document.

Flex possède aussi plusieurs outils lui permettant de nombreuses fonctionnalités très intéressantes. Voici la présentation de ces outils.

### Un outil de Flex : « ProData Database Interface »

C'est grâce à l'interface « ProData » [19] qu'il est possible, à partir de Flex, de se connecter à une base de données.

ProData supporte les systèmes de gestion de base de données suivants :

- Access
- SQL Server
- Oracle
- DB2
- dBase

- Excel

L'interface ProData consiste en deux fichiers : DBLINK.PC et LPADB.W.DLL. Le dernier étant la DLL qui contient le code exécutable qui permet les accès aux systèmes de gestion de base de données via un lien ODBC.

L'interface ProData gère les accès concurrents à une base de données. De plus, aucun téléchargement de la base de données n'est nécessaire. Une fois la connexion faite à une base de données, il est possible d'accéder directement aux données autorisées, ce qui est beaucoup plus rapide. Les lectures et écritures à la base de données peuvent se faire de manière conventionnelle, c'est-à-dire par requête SQL.

#### Un autre outil de Flex : « Intelligence Server »

Intelligence Server [18] est une interface qui permet d'utiliser Win-Prolog et par conséquent Flex, très facilement, à partir d'autres applications écrites dans divers langages de programmation comme C++, Java, Visual Basic, etc.

Voici plusieurs aspects importants et avantageux de l'utilisation de cette interface.

- *Indépendance du langage.* En effet, la composante Win-Prolog d'une application (base de connaissances Flex...) est complètement indépendante du langage de programmation choisi pour l'implémentation de l'application qui utilisera cette composante Win-Prolog avec l'interface Intelligence Server. C'est-à-dire que l'application externe qui utilise une base de connaissances et l'inférence Flex pourrait être codée par exemple en C++, en Java ou en Visual Basic, et ce, sans aucun changement du côté de la base de connaissances et de l'inférence Flex.
- *Multi-utilisateurs.* Il est possible de faire la création de plusieurs instances Intelligence Server à partir d'une même application. C'est-à-dire, par exemple, dans le cas où l'on aurait une application distribuée accessible par plusieurs utilisateurs en même temps, il serait possible de gérer ces accès simultanés à la composante Win-Prolog (exemple une base de connaissances Flex) par l'interface Intelligence Server.
- *Extensibilité.* Comme il a été mentionné précédemment, Intelligence Server peut être utilisé par n'importe quel langage de programmation supporté par Windows et qui supporte l'utilisation de DLL 32-bits.
- *Bonne interaction client-serveur.* Lorsque l'application cliente appelle par exemple Flex pour un raisonnement à l'aide de l'interface Intelligence Server, l'engin d'inférence Flex a la capacité de s'arrêter un moment pour demander à



l'application cliente de lui fournir des informations additionnelles pour qu'il puisse poursuivre son raisonnement.

- *Facilité et simplicité d'utilisation.* L'interface Intelligence Server est très facile d'utilisation et c'est ce que nous allons regarder un peu plus en détail dans la suite de cette section.

Comme il a été mentionné au dernier point, l'interface Intelligence Server est très facile et simple d'utilisation. En effet, au lieu de faire le « mapping » complet entre Prolog et les autres langages de programmation, l'interface Intelligence Server fournit une simple interface texte uniforme à tous les langages de programmation. Le principe général est que l'application cliente envoie ses requêtes à Prolog, ce dernier les exécute par lui-même et renvoi les résultats à l'application cliente par la suite sous forme texte. Pour rajouter à sa simplicité, l'utilisation de l'interface Intelligence Server consiste en l'appel possible de 6 fonctions seulement. Ces fonctions sont définies dans la librairie INT386W.DLL et implémentées dans le fichier INT386W.OVL. Il existe trois versions de cette DLL. Une supportant les langages C, C++ et Delphi, une deuxième supportant le langage Java et une dernière supportant le langage Visual Basic.

Si on résume, la structure d'une application utilisant l'interface Intelligence Server se décompose en 3 parties :

- L'application cliente codée dans un langage supportant l'appel de DLL 32-bits tels que C, C++, Java, Delphi et Visual Basic.
- La partie Prolog de l'application. Par exemple, une base de connaissances Flex.
- L'interface Intelligence Server. Cette interface consiste en deux fichiers, INT386W.DLL et INT386W.OVL, qui définissent et implémentent 6 fonctions. Plus spécifiquement, pour ce qui est des applications Java, les fichiers JINT386W.DLL (interface java dll du fichier int386w.dll) et INT386W.java (classe java interface de jint386w.dll) doivent aussi être utilisés. Voyons plus en détail ces 6 fonctions dans la suite de cette section.

Après étude du logiciel Flex, le seul désavantage noté est qu'il n'est pas aussi convivial que d'autres logiciels, comme Protégé, au niveau de l'éditeur de la base de connaissances. Il s'agit d'une interface du style « console » dans laquelle on doit éditer la base de connaissances dans un fichier texte. Malgré ce désavantage, notre choix s'est arrêté sur ce logiciel pour la construction de notre base de connaissances pour les multiples fonctionnalités et avantages qui viennent d'être présentés.

Avant de poursuivre en détail sur la représentation de la base de connaissances pour le restant de cette section, voyons en quelques lignes les deux autres étapes d'ingénieries des connaissances. Pour ce qui est de l'étape d'**inférence**, qui consiste à faire l'utilisation d'un moteur d'inférence sur la base de connaissances, elle a en grande partie été expliquée précédemment dans la présentation de l'outil Flex. Comme nous l'avons déjà présenté, Flex possède son propre moteur d'inférence et ce dernier sera appliqué sur les règles écrites lors de l'étape de représentation et de saisie des connaissances. De plus, le moteur d'inférence de Flex permet l'utilisation du chaînage avant et du chaînage arrière.

Pour ce qui est de la dernière étape du processus d'ingénierie de la connaissance, celle d'**explications et de justification**, elle n'a pas été appliquée, et ce pour une bonne raison. Le système PDGII exécute un diagnostic de manière autonome. De plus, il n'y a aucune interaction nécessaire entre l'utilisateur et le système durant le diagnostic. Il n'a donc pas été jugé important d'ajouter des éléments d'explications et de justification aux règles de la base de connaissances.

Présentons maintenant en détail la représentation de la nouvelle base de connaissances développée pour le système PDGII.

### Une représentation et une structure flexible et générique

Une fois le logiciel Flex trouvé, il était ensuite temps de faire la représentation ainsi que la saisie des connaissances acquises et validées dans la base de connaissances. Comme une limite majeure du système PDG était que les éléments d'expertise étaient peu flexibles et difficilement modifiables, nous nous devions de structurer les connaissances de manière à ce qu'elles soient plus flexibles. De plus, l'idée derrière la base de connaissances était qu'elle puisse servir, non seulement dans le contexte du système PDGII complet, mais aussi dans différents contextes de recherche sur la performance d'entreprise. La solution était donc de rendre la structure de la base de connaissances la plus générique possible pour que cette dernière soit la plus facilement utilisable, non seulement par le nouveau système PDGII, mais aussi par d'autres systèmes de diagnostic qui pourraient éventuellement être construits par le laboratoire. Par conséquent, un important travail de représentation et de structuration de l'expertise a été accompli [2].

Au cours des années, le laboratoire s'est forgé une large et solide expertise concernant le diagnostic de la performance d'entreprise. Comme nous l'avons vu dans la phase d'acquisition des connaissances (chapitre 4.2.1.1), cette expertise s'est traduite par le développement d'un grand nombre de critères d'évaluation concernant plusieurs domaines de performance d'entreprise. Ces critères d'évaluation constituent donc le cœur de ce que contient la nouvelle base de connaissances. Comme nous voulions que ces critères d'évaluation, et par conséquent la base de connaissances, soient utilisables

par plusieurs systèmes dont le nouveau système PDGII, il fallait représenter ces critères de manière très flexible et générique dans la base de connaissances.

Par conséquent, nous avons organisé et structuré la base de connaissances en deux parties distinctes l'une de l'autre [2]. La première partie (la partie de gauche sur la figure 4.2) contient tous les critères d'évaluation. Étant donné que les critères d'évaluation ne sont pas seulement spécifiques au système PDGII, il s'agit de la partie générique qui peut être utilisée par tous les systèmes qui voudraient utiliser la base de connaissances. On compte en tout 64 critères d'évaluation. La deuxième partie (la partie de droite sur la figure 4.2) constitue la partie spécifique à chaque système. Par conséquent, toutes les connaissances qui sont spécifiques au système PDGII se retrouveront dans cette partie de la base de connaissances.

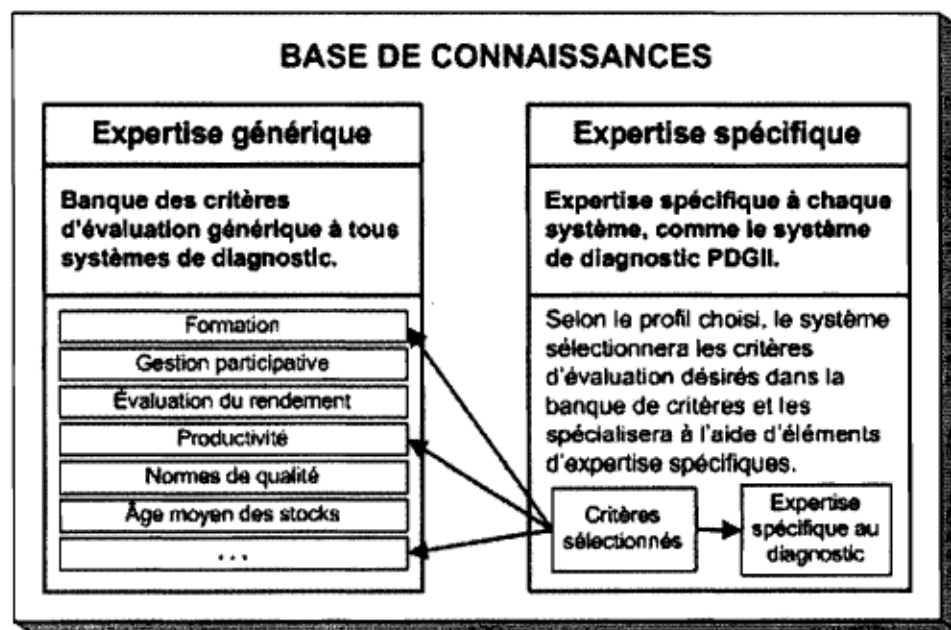


Figure 4.2 Architecture de la base de connaissances

Une limite que représente le système PDG par rapport aux besoins actuels et futurs, est qu'il permet de produire un seul profil de diagnostic, c'est-à-dire le diagnostic complet des 7 secteurs d'évaluation (fonctions et section d'intégration) énumérés précédemment :

- Ressources humaines
- Systèmes de production
- Gestion de la production
- Ventes et marketing
- Gestion et contrôle
- Efficacité

- Vulnérabilité

La structure de la base de connaissances viendra corriger ce problème de flexibilité en permettant la production facile de différents profils de diagnostic dans le futur, selon les besoins du laboratoire. La partie spécifique de la base de connaissances peut donc contenir différents profils [2], par exemple un profil spécialisant le diagnostic sur les secteurs des systèmes de production et de la gestion de la production. Un autre profil pourrait se spécialiser seulement sur le diagnostic des ressources humaines, et ainsi de suite (il est important de mentionner que de nouveaux profils spécialisés nécessiteraient la conception de nouveaux questionnaires spécialisés). Bien entendu, tout comme dans le système PDG, le nouveau système PDGII offrirait le profil du diagnostic complet contenant tous les secteurs d'évaluation de l'entreprise : l'objectif de base de ces deux systèmes.

Un autre concept important dans l'organisation de la base de connaissances réside dans le lien qui existe entre la partie générique et la partie spécifique de cette dernière [2]. En effet, chaque profil contenu dans la base de connaissances est défini par les critères d'évaluation nécessaires pour effectuer le diagnostic associé. Pour se faire et comme nous l'avons vu, chaque profil est défini dans la partie spécifique de la base de connaissances. Ensuite, pour chaque profil, il y a un lien vers la partie générique (la banque de critères d'évaluation) de la base de connaissances pour la sélection de tous les critères d'évaluation dont a besoin le profil en question. Une fois sélectionnés et rendus dans la partie spécifique, les différents critères sont alors adaptés au profil qui les requiert. Ce lien est illustré dans la figure 4.2.

Globalement, cette manière de fonctionner permet donc d'avoir une partie générique contenant tous les critères d'évaluation génériques à tous systèmes éventuels utilisant la base de connaissances, comme le système PDGII. En effet, ces critères d'évaluation contiennent les connaissances générales qui peuvent être utilisées par n'importe quel profil de n'importe quel système. C'est ensuite à un profil de diagnostic (dans la partie spécifique) de sélectionner les critères d'évaluation nécessaires (dans la partie générique) et de les spécialiser comme il le souhaite et comme il en a besoin pour son diagnostic (dans la partie spécifique).

### Une formalisation en trois niveaux d'expertise

Après cette séparation en deux parties de la base de connaissances, une autre notion et amélioration qu'apporte la base de connaissances est la formalisation de son expertise en 3 niveaux différents [2].

Afin de mieux comprendre cette formalisation en 3 niveaux, regardons tout d'abord la structure même d'un diagnostic du système PDGII. Comme nous l'avons vu

précédemment, le système PDGII exécute un diagnostic sur les pratiques de gestion et de production de l'entreprise ainsi que sur ses résultats, et ce, sur différents secteurs d'activités. Chaque secteur d'activité (d'un profil de diagnostic) est lié à plusieurs critères d'évaluation afin d'exécuter un diagnostic spécifique à son secteur. Ces critères, organisés en une banque de critères, constituent la partie générique de la base de connaissances (voir figure 4.2).

Voyons maintenant la structure du diagnostic PDGII ainsi que les 3 niveaux d'expertise reliés (figure 4.3).

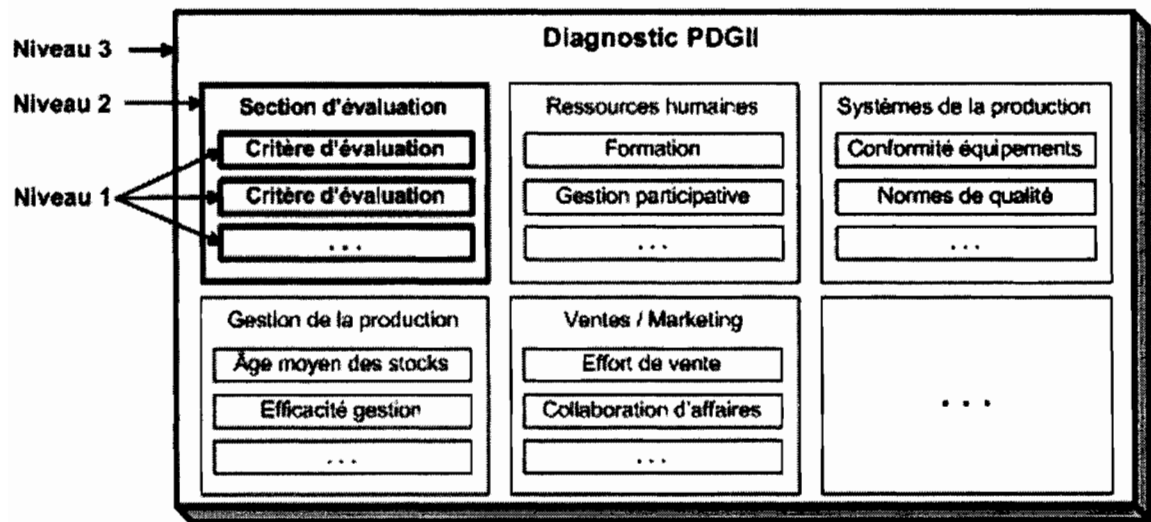


Figure 4.3 Formalisation en 3 niveaux des connaissances du système PDGII

Le **premier niveau** d'expertise correspond à toute l'expertise des critères d'évaluation qui ont été sélectionnés, par un profil du système PDGII, dans la partie générique de la base de connaissances et ensuite spécialisés dans la partie spécifique de cette dernière, comme il a été expliqué précédemment. Parmi les éléments d'expertise dont ont besoin les critères d'évaluation, on retrouve les données de l'entreprise ainsi que celles des entreprises faisant partie du groupe témoin. On retrouve aussi toutes les règles spécifiques pour l'évaluation de ces critères sélectionnés. De plus, on retrouve dans ce premier niveau d'expertise toutes les règles nécessaires à la production des commentaires de recommandation reliés au diagnostic de la performance de ces critères d'évaluation. Bref, ce premier niveau d'expertise contient tous les éléments d'expertise nécessaires à l'évaluation individuelle de tous les critères sélectionnés, et ce pour chaque secteur d'activité, et bien entendu selon le profil choisi par l'utilisateur.

Le **deuxième niveau** d'expertise correspond à toutes les connaissances permettant de faire l'évaluation globale de chaque secteur d'activité d'un profil choisi. Pour ce faire, tous les critères d'évaluation de chaque secteur sont regroupés ensemble pour une

évaluation globale. La pondération de chaque critère d'évaluation par rapport à son secteur d'activité est l'élément d'expertise jouant un rôle central dans cette évaluation sectorielle. Nous retrouvons aussi dans ce niveau d'expertise, toutes les règles permettant le diagnostic global de la performance de chaque secteur, et par conséquent, les règles permettant l'identification des critères d'évaluation présentant la plus faible performance. C'est à l'aide de ces critères faibles identifiés et de l'expertise du premier niveau que le système peut ainsi présenter des commentaires de recommandation sur la performance de l'entreprise pour chaque secteur d'activité.

Le dernier et **troisième niveau** d'expertise est associé à l'évaluation globale de tous les secteurs d'activités correspondant au profil choisi. Il s'agit donc de l'évaluation globale de la performance de l'entreprise. Ce niveau d'expertise est le plus complexe à représenter puisqu'il demande la comparaison entre les secteurs d'activités afin d'attribuer des poids ou une importance relative à ceux-ci. Cette pondération d'un secteur par rapport à un autre est une expertise que le laboratoire ne possède pas encore complètement. Comme le modèle d'expertise de ce troisième niveau n'est pas complet, il ne peut être utilisé pour le moment par le système PDGII. Par conséquent, cette évaluation globale du diagnostic est faite manuellement par un expert du laboratoire. Le modèle du troisième niveau d'expertise est présentement en développement par les experts du laboratoire et sera ajouté éventuellement à la base de connaissances.

#### **4.2.1.2 Contenu**

Après avoir vu la représentation à un niveau de détail plus abstrait et conceptuel, nous allons maintenant, dans cette section, détailler le contenu de cette représentation de la connaissance à un niveau plus concret. Bien entendu, par souci de confidentialité de l'expertise, le contenu de la base de connaissances ne sera pas donné dans ses détails. Il sera question ici de donner un niveau plus concret de la représentation de la connaissance, en y présentant quelques exemples à l'appui.

##### Partie générique – Critères d'évaluation

Comme nous l'avons vu précédemment, la base de connaissances se sépare en deux parties principales. La première partie est celle de l'expertise générique à tous systèmes du domaine dans laquelle on retrouve tous les critères d'évaluation de la performance d'entreprise. Ces critères d'évaluation sont définis par plusieurs informations comme nous pouvons le voir en détail dans le tableau 4.1.

Attributs du frame d'un critère	Type de l'attribut
Nom	Chaîne de caractères
Description	Chaîne de caractères
Type	Choix entre « pratique » ou « résultat »
Variables de base de données	Liste de chaîne de caractères
Concept de benchmarking associés (il peut y avoir de 1 à N de cet attribut)	Frame « benchmark_data » décrivant des données de benchmarking.
Attributs du frame « benchmark_data »	Type de l'attribut
Valeur de l'entreprise	Valeur numérique
Valeur du groupe témoin	Valeur numérique
Score de comparaison (benchmark)	Valeur numérique entre 0 et 4
<b>Exemple</b>	
<pre> frame critere_evaluation .  frame formation is a critere_evaluation ;     default nom          is  'Formation'          and     default description   is  'Description du critère...' and     default type_critere  is  pratique              and     default variables_BD  are  {BUFOEMTOS01,BUFOEMTOA00} and     default annee_derniere is a benchmark_data      and     default annee_courante is a benchmark_data .  frame benchmark_data ;     default valeur_entreprise is _ and     default valeur_gt         is _ and     default score_comparaison is _ . </pre>	

Tableau 4.1 Description du frame d'un critère d'évaluation et du frame « benchmark\_data ».

Dans le tableau 4.1, on retrouve l'exemple d'un frame d'un critère d'évaluation faisant partie de la banque de critères de la partie générique de la base de connaissances. Il s'agit du frame du critère d'évaluation « formation » qui peut servir à l'évaluation du niveau de formation des ressources humaines dans l'entreprise.

L'attribut « type » de ce critère est de type « pratique », c'est-à-dire qu'il évalue une pratique de l'entreprise. Les critères de type « résultats » quant à eux évaluent les résultats de l'entreprise sur un aspect donné.

L'attribut « variables\_BD » contient les noms des variables de la base de données du laboratoire dans lesquelles on retrouve les données des concepts de ce critère d'évaluation. Ces concepts sont définis par tous les attributs restants du frame. Bien entendu, le nombre d'attributs définissant les concepts d'un critère est toujours égal au nombre de variables contenues dans l'attribut « variables\_BD ». Chaque concept possède donc sa variable dans la base de données. Pour ce qui est du type de ces attributs

concepts, ils sont du type du frame « benchmark\_data ». Ce frame, qui est aussi défini dans l'exemple ci-dessus du tableau 4.1, sert à représenter toutes les données qui serviront à la méthode de « benchmarking » [25, 33] utilisée par les critères pour l'évaluation de la performance d'entreprise.

Le frame « benchmark\_data » contient 3 attributs. Les deux premiers attributs sont ceux des valeurs respectives de l'entreprise et du groupe témoin concernant un des attributs concepts (une des variables) d'un critère donné. Le troisième attribut correspond au score de comparaison de la valeur d'entreprise et celle du groupe témoin, calculé à l'aide des règles de « benchmarking » implémentées dans la base de connaissances.

Bref, comme nous pouvons le voir à la figure 4.4, cette partie d'expertise générique de la base de connaissances consiste essentiellement en une banque de critères d'évaluation générique à tous systèmes dont un attribut est de type du frame « benchmark\_data ».

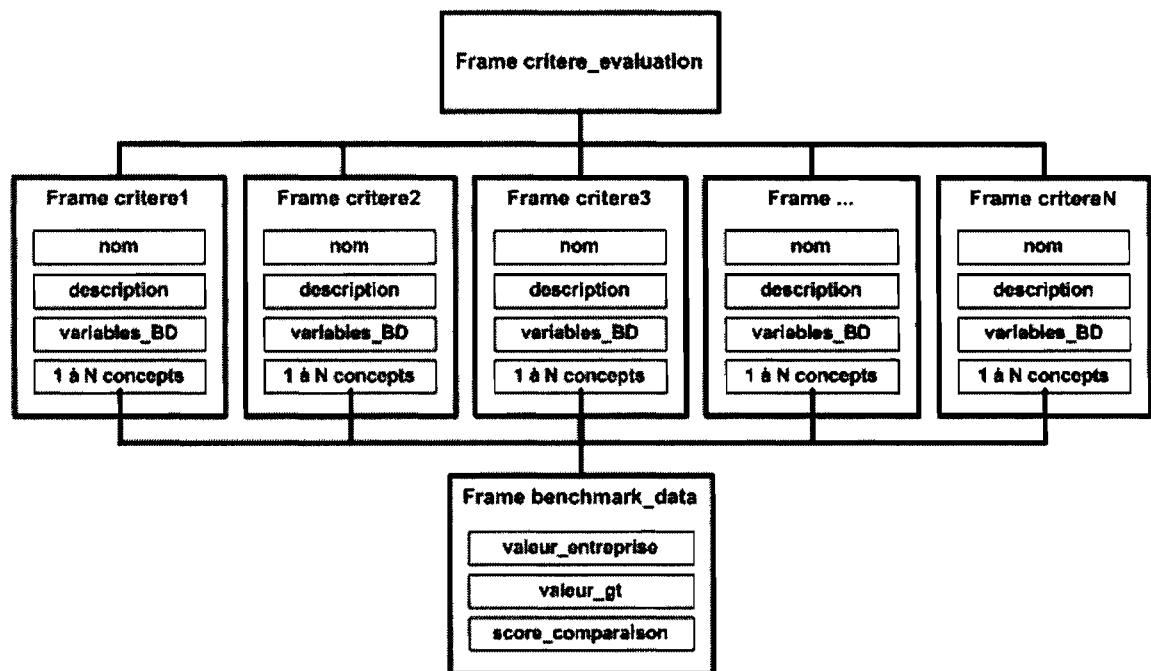


Figure 4.4 Partie générique de la base de connaissances (banque de critères)

### Partie spécifique

Maintenant, voyons plus concrètement en quoi consiste la partie spécifique de l'expertise de la base de connaissances. Il s'agit de tous les éléments d'expertise qui sont



spécifiques aux différents systèmes utilisant la base de connaissances. Pour l'instant, seulement le système PDGII utilise la base de connaissances.

La partie spécifique de l'expertise du système PDGII contient d'abord le frame « **global** » (tableau 4.2), qui renferme des attributs servant au fonctionnement général du diagnostic PDGII. Ces attributs sont utilisés dans les règles et procédures du diagnostic. La plupart de ces attributs sont configurés par l'utilisateur au début de la production d'un diagnostic via le programme utilisateur.

Attributs du frame global	Type de l'attribut	Description
g_profil	Instance du frame profil	Profil de diagnostic choisi.
g_connexionBD	Chaîne de caractères	Pour la connexion à la base de données.
g_tableBD	Chaîne de caractères	Table de la base de données dans laquelle se trouvent les données pour le diagnostic.
g_serieBD1	Chaîne de caractères	Identificateur de la 1ère série de données (entreprise ou groupe témoin).
g_serieBD2	Chaîne de caractères	Identificateur de la 1ère série de données (entreprise ou groupe témoin).
g_noEssai	Valeur numérique	Numéro de l'essai de ce diagnostic. Plusieurs essais sont possibles pour un diagnostic.
g_benchmarkDataType	Chaîne de caractères	Type de données benchmark. Nom du frame benchmark (ex : benchmark_data).
g_nbRangFaiblessePratique	Valeur numérique	Nombre de critères de type pratiques à identifier comme faibles pour chaque section évaluée.
g_nbRangFaiblesseResultat	Valeur numérique	Nombre de critères de type résultats à identifier comme faibles pour chaque section évaluée.
g_nbAnneesCalculHisto	Valeur numérique	Nombre d'années à prendre en compte pour le calcul historique.
g_langue	Chaîne de caractères	Spécifie la langue du diagnostic pour les éléments qui se retrouveront dans le

		rapport de diagnostic. qc = français du Québec fr = français de France en = anglais
--	--	--

**Tableau 4.2 Description du frame « global » .**

Dans la description de ce frame, il est question du calcul historique. Le calcul historique est un type de calcul réservé à certains critères d'évaluation. Des calculs différents doivent être exécutés pour faire l'évaluation de la performance de l'entreprise pour ces critères d'évaluation. Des calculs de tendance, de signe et d'évolution doivent être exécutés pour effectuer le calcul historique.

Le calcul de tendance sert à évaluer la performance de l'entreprise par rapport à elle-même au courant des années. Par exemple, on pourrait dire qu'il y a eu une croissance concernant un critère d'évaluation sur les ventes de l'entreprise (l'entreprise vend plus cette année que l'an passé).

Le calcul de signe sert à faire des comparaisons ponctuelles entre l'entreprise et son groupe témoin, et ce pour chaque année. Par exemple, l'an passé, l'entreprise a eu une plus faible croissance de ses ventes que son groupe témoin, et cette année, elle a eu la même croissance des ventes que son GT.

Le calcul d'évolution sert à faire la comparaison globale des signes, qui indique si l'entreprise s'est améliorée dans le temps par rapport à son groupe témoin. Par exemple, l'an passé, l'entreprise avait une croissance de 10% de ses ventes, et cette année 2%. Mais le groupe témoin est à 3% depuis les 3 dernières années. Donc l'entreprise ne se dirige pas dans la bonne direction.

Ensuite, il y a le frame « **profil** » et toutes ses instances. Ce frame sert à définir en quoi consiste un profil de diagnostic. Il s'agit d'un frame de base et chaque instance de ce frame définit un profil spécifique de diagnostic. Voici dans le tableau 4.3, la description de ce frame ainsi qu'un exemple d'une de ces instances « profil\_complet »

Attributs du frame profil	Type de l'attribut	Description
Nom	Chaîne de caractères	Nom du profil.
Description	Chaîne de caractères	Description du profil.
Sections	Liste d'instance du type frame « section ».	Liste qui contient tous les noms d'instance du frame « section » qui font partie de ce profil.
<b>Exemple d'instance</b>		

```

frame profil ;
    default nom          is 'Donner un nom au profil...' and
    default description  is 'Description au profil...'  and
    default sections     are {} .

instance profil_complet is a profil ;
    nom          is 'Profil complet' and
    description  is 'Profil complet du diagnostic PDGII.' and
    sections     are { section_systemesProduction
                      , section_gestionProduction
                      , section_ressourcesHumaines
                      , section_ventesMarketing
                      , section_gestionControle
                      , section_efficacite
                      , section_vulnerabilite
                      } .

```

**Tableau 4.3 Description du frame « profil » .**

Pour ce qui est du frame « **section** » (tableau 4.4), il s'agit aussi d'un frame de base avec lequel on doit créer des instances. Chaque instance de ce frame correspond à une section d'évaluation dans un diagnostic. Par exemple, on peut avoir la section d'évaluation des ressources humaines, la section d'évaluation des systèmes de production, etc. Une ou plusieurs instances de sections d'évaluation feront partie d'une instance du frame « profil » comme on a pu le voir dans le tableau 4.3. Voici le tableau 4.4 qui décrit le frame « section » et un exemple d'instance.

Attributs du frame section	Type de l'attribut	Description
Nom	Chaîne de caractères	Nom de la section.
Description	Chaîne de caractères	Description de la section.
total_scores_pondere	Valeur numérique	Total des scores pondérés de tous les critères d'évaluation faisant partie de cette section.
total_ponderation	Valeur numérique	Total des pondérations de tous les critères d'évaluation faisant partie de cette section.
score_section	Valeur numérique	Score que la section a obtenu lors de l'évaluation globale de la section.
commentaire_pratique	Chaîne de caractères	Commentaire de recommandation généré par le système sur les pratiques de l'entreprise.
commentaire_resultat	Chaîne de caractères	Commentaire de recommandation généré par le

		système sur les résultats de l'entreprise.
<b>Exemple d'instance</b>		
<pre> frame section   default nom is 'Nommer la section...' and   default description is 'Décrire la section...' and   default total_scores_ponderes is 0 and   default total_ponderation is 0 and   default score_section is 0 and   default commentaire_pratique is nothing and   default commentaire_resultat is nothing .  instance section_systemesProduction is a section ;   nom is 'Systèmes de production' and   description is 'Section de profil qui comprend les critères pour     l''évaluation des systèmes de production.' . </pre>		

**Tableau 4.4 Description du frame « section » .**

Rappelons-nous le fonctionnement des deux parties de la base de connaissances. Nous avons la partie générique qui contient, comme nous l'avons vu tous les critères d'évaluation génériques à tous systèmes. La partie spécifique quant à elle contient l'expertise spécifique de systèmes, par exemple le système PDGII. Cette expertise du système PDGII dans la partie spécifique crée un lien avec la partie générique afin de sélectionner tous les critères d'évaluation génériques dont il a besoin pour son diagnostic. Une fois cette sélection effectuée, le système PDGII spécialise les critères qu'il a sélectionnés à l'aide de connaissances plus spécifiques au diagnostic PDGII (voir chapitre 4.2.1.1).

Cette spécialisation débute à l'aide du frame « **critere\_PDGII** ». Ce frame servira de frame de base pour tous les critères sélectionnés par le système PDGII. Une instance de ce frame sera créée pour chaque critère sélectionné. Les valeurs de la plupart des attributs de ses instances seront données au fur et à mesure du diagnostic. Voici la description de ce frame dans le tableau 4.5.

Attributs du frame <b>critere_PDGII</b>	Type de l'attribut	Description
type_critere	Chaîne de caractères	Type de critères (pratique ou résultat). Cette valeur proviendra du critère sélectionné.
Section_utilisation	Chaîne de caractères	Section dans laquelle sera utilisé le critère sélectionné.
Ponderation	Valeur numérique	Valeur de pondération du critère

		sélectionné par rapport aux autres critères de sa section.
score_criter	Valeur numérique	Score que le critère obtiendra lors du diagnostic.
score_pondere	Valeur numérique	Score pondéré que le critère obtiendra lors du diagnostic. (score_criter * ponderation)
score_pondere_inv	Valeur numérique	Score pondéré inverse que le critère obtiendra lors du diagnostic. (score_criter * 1/ponderation)
rang_faiblesse	Valeur numérique	Rang de faiblesse du critère dans sa section. (1 signifiant que le critère est le plus faible de sa section.
Commentaire	Chaîne de caractères	Commentaire de recommandation produit par le diagnostic sur ce critère s'il y a lieu.
commentaire_tendance	Chaîne de caractères	Commentaire de recommandation par rapport à la tendance (calcul historique) produit par le diagnostic sur ce critère s'il y a lieu.
commentaire_signe	Chaîne de caractères	Commentaire de recommandation par rapport au signe (calcul historique) produit par le diagnostic sur ce critère s'il y a lieu.
commentaire_evolution	Chaîne de caractères	Commentaire de recommandation par rapport à l'évolution (calcul historique) produit par le diagnostic sur ce critère s'il y a lieu.
<pre> frame critere_PDGI ;   default type_critere      is nothing and   default section_utilisation is _      and   default ponderation       is _      and   default score_critere     is -200    and   default score_pondere     is -200    and   default score_pondere_inv is -200    and   default rang_faiblesse    is _      and   default commentaire       is _      and </pre>		

<b>default</b>	commentaire_tendance	<b>is</b>	nothing	<b>and</b>
<b>default</b>	commentaire_signe	<b>is</b>	nothing	<b>and</b>
<b>default</b>	commentaire_evolution	<b>is</b>	nothing	<b>.</b>

**Tableau 4.5 Description du frame « critere\_PDGI » .**

Pour faire suite au frame de base « critere\_PDGI » qui vient d’être présenté, voyons maintenant le frame « **critere\_selectionne** » qui se trouve à être un frame enfant du frame « critere\_PDGI » et qui sert à la création des instances pour tous les critères sélectionnés. Le tableau 4.6 présente la description de ce frame et un exemple de critères sélectionnés par la création d’instance de ce frame.

<b>Attributs du frame critere_selectionne</b>	<b>Type de l’attribut</b>	<b>Description</b>
nomCritere	Frame d’un critère présent dans la banque de critères de la partie générique.	Identifie le frame du critère sélectionné dans la banque de critères de la partie générique de la base de connaissances.
methodeCalculScoreCritere	Chaîne de caractères	Correspond à la méthode qui sera utilisée pour le calcul du score du critère.
calculBenchmark	Chaîne de caractères	Correspond à la règle qui sera utilisée pour le calcul benchmark (score de comparaison entre l’entreprise et le groupe témoin pour chaque concept ou variable d’un critère, voir tableau 4.1).
ponderation_tendance	Valeur numérique	Pondération pour le calcul de la tendance dans le cas d’un critère historique.
ponderation_signe	Valeur numérique	Pondération pour le calcul du signe dans le cas d’un critère historique.
ponderation_evolution	Valeur numérique	Pondération pour le calcul de l’évolution dans le cas d’un critère historique.
seuil_faible	Valeur numérique	Seuil faible servant dans les règles de diagnostic. Aide à l’identification des intervalles de performance.
seuil_fort	Valeur numérique	Seuil fort servant dans les règles de diagnostic. Aide à

		l'identification des intervalles de performance.
seuil_tendance	Valeur numérique	Seuil de tendance servant dans les règles de diagnostic dans le cas de critères historiques.
seuil_signe	Valeur numérique	Seuil du signe servant dans les règles de diagnostic dans le cas de critères historiques.
seuil_evolution	Valeur numérique	Seuil de l'évolution servant dans les règles de diagnostic dans le cas de critères historiques.
Sens	Valeur numérique	Le sens est une valeur égale à 0 ou à 1 et sert à déterminer dans quel sens se fait l'interprétation d'un score de performance. Une valeur pour l'entreprise au-dessus du groupe témoin peut être positif (sens = 0) ou négatif (sens = 1), selon le critère.
score_tendance	Valeur numérique	Score de tendance dans le cas d'un critère historique.
score_signe	Valeur numérique	Score de signe dans le cas d'un critère historique.
score_evolution	Valeur numérique	Score de l'évolution dans le cas d'un critère historique.

#### Exemples d'instance

```

frame critere_selectionne is a critere_PDGUII ;
    default nomCritere is _ and
    default methodeCalculScoreCritere is calculOrdinaire and
    default calculBenchmark is eval_score and
    default ponderation_tendance is _ and
    default ponderation_signe is _ and
    default ponderation_evolution is _ and
    default seuil_faible is 0.1 and
    default seuil_fort is 0.5 and
    default seuil_tendance is 0.05 and
    default seuil_signe is 0.1 and
    default seuil_evolution is 0.1 and
    default sens is _ and
    default score_tendance is -102 and
    default score_signe is -103 and
    default score_evolution is -104 .

instance critere29 is a critere_selectionne ;
    nomCritere is formation and
    ponderation is 0.12 and

```

sens	is	0	and
commentaire	is	commentaire_formation	and
section_utilisation	is	section_ressourcesHumaines	.
<b>instance critere26a is a critere_selectionne ;</b>			
nomCritere	is	age_moyen_comptes_clients	and
ponderation	is	0.09	and
sens	is	1	and
methodeCalculScoreCritere	is	calculHistorique	and
ponderation_tendance	is	0.25	and
ponderation_signe	is	0.56	and
ponderation_evolution	is	0.19	and
commentaire	is	commentaire_age_moyen_comptes_clients	and
section_utilisation	is	section_gestionControle	.

**Tableau 4.6 Description du frame « critere\_selectionne » .**

Un autre frame est lié aux derniers frames présentés. Il s'agit du frame « **jointure\_criteres\_selectionnes** », enfant du frame « critere\_selectionne ». Grâce à ce frame, il est possible de créer des instances permettant le regroupement de critères sélectionnés afin de faire une évaluation conjointe de ces critères. À l'aide de règles, certains critères, selon leur résultat de performance, peuvent être évalués individuellement ou bien en groupe avec d'autres critères. Afin de bien identifier ces critères parmi les connaissances du système PDGII, il suffit de créer des instances du frame « jointure\_criteres\_selectionnes », et de spécifier dans l'attribut « criteres » les différentes instances de critères « critere\_selectionne » qui peuvent former une jointure.

Par exemple, on peut avoir deux critères :

- Le critere34a et le critere34b qui sont deux critères instances de type frame « critere\_selectionne ».
- Ces deux critères sont indépendants comme tous les autres critères sélectionnés.

Dans certains cas (selon les règles), ces deux critères peuvent être évalués ensemble :

- On utilise donc l'instance du frame « jointure\_criteres\_selectionnes » qui est associé avec la valeur {critere34a, critere34b} pour l'attribut « criteres ». Voir l'exemple dans le tableau 4.7.

On retrouve donc une instance du frame « jointure\_criteres\_selectionnes » pour chaque groupe de critères nécessaire. Ces groupes font partie de l'expertise et ont par conséquent été identifiés par l'expert humain. Voici dans le tableau 4.7, la description de ce frame ainsi qu'un exemple d'instance.



Attributs du frame <b>jointure_criteres_selectionnes</b>	Type de l'attribut	Description
Criteres	Liste d'instances de type frame critere_selectionne.	Liste identifiant tous les critères pouvant être joints pour une évaluation conjointe.
<b>Exemples d'instance</b>		
<pre> frame jointure_criteres_selectionnes is a critere_selectionne ;   default criteres          is {} .  instance jointure_taux_departs_volontaires is a                                      jointure_criteres_selectionnes ;    criteres          are {critere34a, critere34b}          and   ponderation       is 0.15                               and   commentaire       is commentaire_jointure_taux_departs_volontaires and   section_utilisation is section_ressourcesHumaines . </pre>		

Tableau 4.7 Description du frame jointure\_criteres\_selectionnes » .

Une autre partie de l'expertise qui se retrouve dans la partie spécifique de la base de connaissances pour le système PDGII correspond à l'expertise des commentaires de recommandation. Une structure de tous les commentaires possibles de tous les critères existants a été construite dans la base de connaissances. Cette structure est constituée de deux frames « **commentaire\_critere\_pratique** » et « **commentaire\_critere\_resultat** » et de leurs instances. Voici le tableau 4.8 décrivant ces deux frames et leurs instances.

Attributs des frame <b>commentaire_critere_pratique</b> <b>Commentaire_critere_resultat</b>	Type de l'attribut	Description
recommandation_qc	Chaîne de caractères.	Recommandation à faire pour ce critère s'il y a lieu (diagnostic français québécois).
recommandation_fr	Chaîne de caractères.	Recommandation à faire pour ce critère s'il y a lieu (diagnostic français européen).
recommandation_en	Chaîne de caractères.	Recommandation à faire pour ce critère s'il y a lieu (diagnostic anglais).

informations_absentes_qc	Chaîne de caractères.	Commentaire de remplacement lorsque des informations sont manquantes (diagnostic français québécois.)
informations_absentes_fr	Chaîne de caractères.	Commentaire de remplacement lorsque des informations sont manquantes (diagnostic français européen.)
informations_absentes_en	Chaîne de caractères.	Commentaire de remplacement lorsque des informations sont manquantes (diagnostic anglais.)

#### Exemples d'instance

```

frame commentaire_critere_pratique
  default recommandation_qc      is 'À court terme, aucune
modification particulière des pratiques utilisées n'est suggérée à
l'entreprise.' and
  default recommandation_fr      is '' and
  default recommandation_en      is '' and
  default informations_absentes_qc is 'Compte tenu de l'absence
de certaines informations, les pratiques de l'entreprise ne peuvent
être évaluées. Les informations du groupe témoin sont tout de même
présentées à titre indicatif.' and
  default informations_absentes_fr is '' and
  default informations_absentes_en is '' .

instance commentaire_formation is a commentaire_critere_pratique
  recommandation_qc is 'Une amélioration de la performance de la
gestion des ressources humaines pourrait être envisagée, notamment par
une augmentation des activités de formation afin de développer les
compétences, l'efficacité et la flexibilité des employés.' .

frame commentaire_critere_resultat
  default recommandation_qc      is 'Vu les résultats obtenus,
aucune recommandation particulière n'est formulée.' and
  default recommandation_fr      is '' and
  default recommandation_en      is '' and
  default informations_absentes_qc is 'Compte tenu de l'absence
de certaines informations, aucun commentaire n'est formulé sur les
résultats obtenus par l'entreprise. Les informations du groupe témoin
sont présentées à titre indicatif.' and
  default informations_absentes_fr is '' and
  default informations_absentes_en is '' .

```

```
instance commentaire_qualite_relations_travail is a
commentaire_critere_resultat
    recommandation_qc is 'Il peut être nécessaire de porter une
attention particulière à la qualité des relations de travail afin de
réduire les tensions et accroître l'efficacité des employés.' .
```

**Tableau 4.8 Description des frames « commentaire\_critere\_pratique » et  
« commentaire\_critere\_resultat ».**

Il est à noter que les commentaires autres que français québécois ne sont pas présents dans les exemples, car au moment d'écrire ces lignes, ils n'avaient pas été saisis dans la base de connaissances.

### Règles et procédures

Jusqu'à présent, nous avons parcouru le contenu de la base de connaissances, autant pour les éléments d'expertise de la partie générique que de la partie spécifique. Comme nous l'avons expliqué précédemment, cette partie spécifique de la base de connaissances spécialise aussi les critères sélectionnés à l'aide de règles et de procédures afin de permettre le diagnostic PDGII. Sans entrer dans les détails, voici un aperçu de ce contenu (règles et procédures).

Le tableau 4.9 présente les différentes étapes d'un diagnostic et les règles qui y sont associées.

#### **ÉTAPE 1 : Création des instances des critères sélectionnés du profil de diagnostic choisi.**

**Règles :** creation\_instances\_criteres\_selectionnes  
eval\_score  
eval\_score2

#### **Procédure : creationInstance**

Sert à faire la création de toutes les instances des critères qui ont été sélectionnés par le profil dans la banque de critères (partie générique). La règle « creation\_instances\_criteres\_selectionnes » identifie chaque critère sélectionné par le profil, et appelle la procédure « creationInstance » pour chacun de ces critères. Cette procédure crée l'instance d'un critère sélectionné, identifie ses attributs benchmark, accède à la base de données pour donner la valeur entreprise et la valeur groupe témoin à chacun de ses attributs, et calcule le score benchmark de comparaison pour chacun de ses attributs à l'aide des règles « eval\_score » ou

« eval\_score2 ».

## **ÉTAPE 2 : Calcul des scores et des scores pondérés des critères sélectionnés.**

**Règles :** calcul\_score\_methode\_ordinaire  
calcul\_score\_methode\_historique  
tendance  
signe  
evolution

**Procédures :** calculOrdinaire  
calculHistorique  
calculer\_tendance  
calculer\_signe  
calculer\_evolution  
calculScoresPonderes

Dans cette étape, tous les critères sélectionnés seront soumis à ces différentes règles et procédures selon qu'il s'agisse de critères ordinaires ou historiques. Plus précisément, ces règles et procédures feront le calcul du score et du score pondéré (par rapport à sa section) de chacun de ces critères du profil.

## **ÉTAPE 3 : Jointure de critères qui le nécessite.**

**Règle :** calcul\_jointure\_criteres

**Procédure :** jointureCriteres

Dans l'évaluation, certains critères peuvent être regroupés ensemble pour faire une évaluation globale de ce groupe de critères. Voici les cas pour lesquels on va regrouper plusieurs critères en un seul (création d'une nouvelle instance):

- si tous les critères du groupe potentiel ont des scores  $< 2$
- si tous les critères du groupe potentiel ont des scores  $> 2$

En cas de jointure, une nouvelle instance est créée et le rang des critères qui composent cette nouvelle instance est mis à 999999 pour être sûr qu'ils ne seront pas pris en compte pour l'identification des critères faibles.

## **ÉTAPE 4 : Identification des critères les plus faibles pour chaque section.**

**Règle :** calcul\_jointure\_criteres

**Procédure :** identificationCriteresFaibles

Sert à identifier les critères pratiques et résultats les plus faibles pour chaque section

du profil. Il est à noter que seuls les critères dont le score est plus petit que 2 sont pris en compte. Le plus faible critère est celui qui possède le plus faible score\_pondere\_inverse. En cas d'égalité entre deux critères, le critère dont la pondération est la plus élevée sera considéré le plus faible.

#### **ÉTAPE 5 : Production des recommandations pour chaque section.**

**Règle :** commentaires\_score\_sections

**Procédure :** creerCommentairesSection

Sert à générer les recommandations des pratiques et des résultats pour chaque section.

**Tableau 4.9 Description des étapes et règles d'un diagnostic.**

D'autres règles et procédures moins directement liées au diagnostic, sont utilisées par la base de connaissances. C'est le cas entre autres dans la configuration du diagnostic et aussi pour la sauvegarde des résultats du diagnostic.

### **4.2.1.3 Solutions apportées**

Le développement d'une base de connaissances pour le nouveau système PDGII est venu résoudre plusieurs limites importantes du système PDG que nous voulions corriger.

#### Représentation centralisée et structurée des connaissances

La base de connaissances a permis la centralisation de tous les éléments de connaissance de manière structurée. Il est donc maintenant beaucoup plus facile :

- De retrouver un élément de connaissance.
- De modifier un élément de connaissance.
- De rajouter un élément de connaissance.
- De supprimer un élément de connaissance.
- De comprendre rapidement dans son ensemble les éléments de connaissance.

#### Formalisation de l'expertise

La formalisation de la base de connaissances en deux parties (générique et spécifique) ainsi que la formalisation de l'expertise du diagnostic PDGII en 3 niveaux sont des

améliorations importantes pour l'avenir de la base de connaissances, et pour l'évolution de l'expertise du LaRePE sur le domaine de la performance des entreprises.

#### Utilisation d'un outil adapté

L'utilisation du logiciel Flex est en grande partie responsable de l'amélioration majeure du nouveau système PDGII. En effet, c'est grâce à cet outil que la base de connaissances, élément central du nouveau système, a pu être construite. De plus, il permet l'abandon du logiciel Microsoft Excel et du langage VB pour un logiciel (Flex) et un langage (KSL) beaucoup plus adaptés au stockage des connaissances (base de connaissances) ainsi qu'au développement d'un système expert de diagnostic (moteur d'inférence).

### **4.2.2 Programme utilisateur**

Un autre élément de solution important du système PDGII est son programme utilisateur. Le programme utilisateur du système PDG était assez difficile d'utilisation et offrait peu de possibilités de configuration du diagnostic. L'objectif premier de ce nouveau programme était d'obtenir une facilité d'utilisation et une configuration rapide du diagnostic.

#### **4.2.2.1 Technologie et présentation du programme**

##### Intégration dans l'environnement informatique du laboratoire

Le LaRePE possède un environnement de travail informatique bien développé. Tous les chercheurs et assistants de recherche ont accès au réseau informatique du laboratoire à partir de leur poste de travail. De plus, un intranet a été développé, permettant à ces mêmes personnes, selon leur droit d'accès, d'accéder aux différents outils de recherche du laboratoire via leur navigateur Web. Cet environnement informatique (intranet, outils) a été développé à l'aide des langages de programmation Java et JSP, ce dernier étant une technologie Web de Java.

Nous avons fait appel à ces mêmes technologies pour le développement du programme utilisateur afin qu'il puisse bien s'intégrer à l'intranet du laboratoire. Étant accessible par un navigateur Web via l'intranet du LaRePE, il n'est installé sur aucun ordinateur du laboratoire, ce qui simplifie grandement les mises à jour du programme.

### Sécurité et contrôle d'accès aux données et à l'expertise

Les données (entrepôt de données) et l'expertise (base de connaissances) sont installées sur un serveur sécurisé dont on peut gérer les droits d'accès via le système de gestion de l'intranet du LaRePE. Cela constitue une grande sécurité pour les données et les connaissances (expertise) du système. De plus, la gestion des droits d'accès constitue une fonction que devait absolument posséder le nouveau système PDGII en raison de la trop grande valeur des données et des connaissances (expertise) du système. Pour ce qui est de l'administration du système, elle en est de beaucoup facilitée.

### Une interface efficace, utile, simple et conviviale

Le programme utilisateur a été pensé et développé afin de donner le plus de contrôle possible à l'utilisateur quant au diagnostic qu'il désire obtenir. Grâce à ce programme, l'utilisateur peut configurer son diagnostic sur plusieurs points. Voici une liste des tâches dont l'utilisateur a le contrôle :

- Choix du profil de diagnostic.
- Choix des critères de sélection pour la création du groupe témoin.
- Choix de la langue désirée pour le rapport du diagnostic.
- Choix de la devise des résultats et des données contenues dans le rapport.
- Modification des pondérations de tous les critères d'évaluation contenus dans le profil de diagnostic choisi.
- Modification des résultats (graphiques, commentaires de recommandation) contenus dans le rapport du diagnostic.

Afin de mieux comprendre l'efficacité, l'utilité, la simplicité et la convivialité du programme utilisateur, voici une séquence de dessins d'écran accompagnée d'explications expliquant la séquence suivie par l'utilisateur pour produire un diagnostic du début à la fin.

L'entrée dans le programme utilisateur est illustrée par la figure 4.5. L'utilisateur doit y choisir le modèle de diagnostic qu'il désire, dans notre cas il s'agit du PDGII. Il doit aussi faire le choix de la langue dans laquelle le diagnostic doit être fait. Actuellement, le diagnostic peut se faire dans trois langues :

- Français (Canada) pour les entreprises francophones du Canada
- Français (Europe) pour les entreprises francophones de l'Europe
- Anglais

Ce choix de langue a une grande importance puisque l'expertise et les profils disponibles pour le diagnostic de la performance d'une entreprise varient selon que l'entreprise vient du Canada ou de l'Europe.

Il est intéressant de noter que tout au long du processus de création d'un diagnostic, l'utilisateur sait toujours où il en est rendu grâce à la liste des étapes qui se retrouve en sous-titre dans chaque interface. Par exemple, dans la figure 4.5, l'utilisateur peut facilement voir qu'il est dans l'étape de choix des paramètres de base.

Choix des paramètres de base

**MODELE DU DIAGNOSTIC**

Choisir parmi les modèles suivants:

☒ PDGII

☐ TestUnitaires

**LANGUE DU DIAGNOSTIC**

Choisir parmi les langues suivantes:

☒ français (Canada)

☐ français

☐ anglais

Figure 4.5 Choix des paramètres de base

Pour faire suite au choix de modèle et de langue, la liste des profils de diagnostic existants est présentée à l'utilisateur tel qu'illustré dans la figure 4.6. Ce dernier n'a qu'à sélectionner celui qu'il désire pour son diagnostic



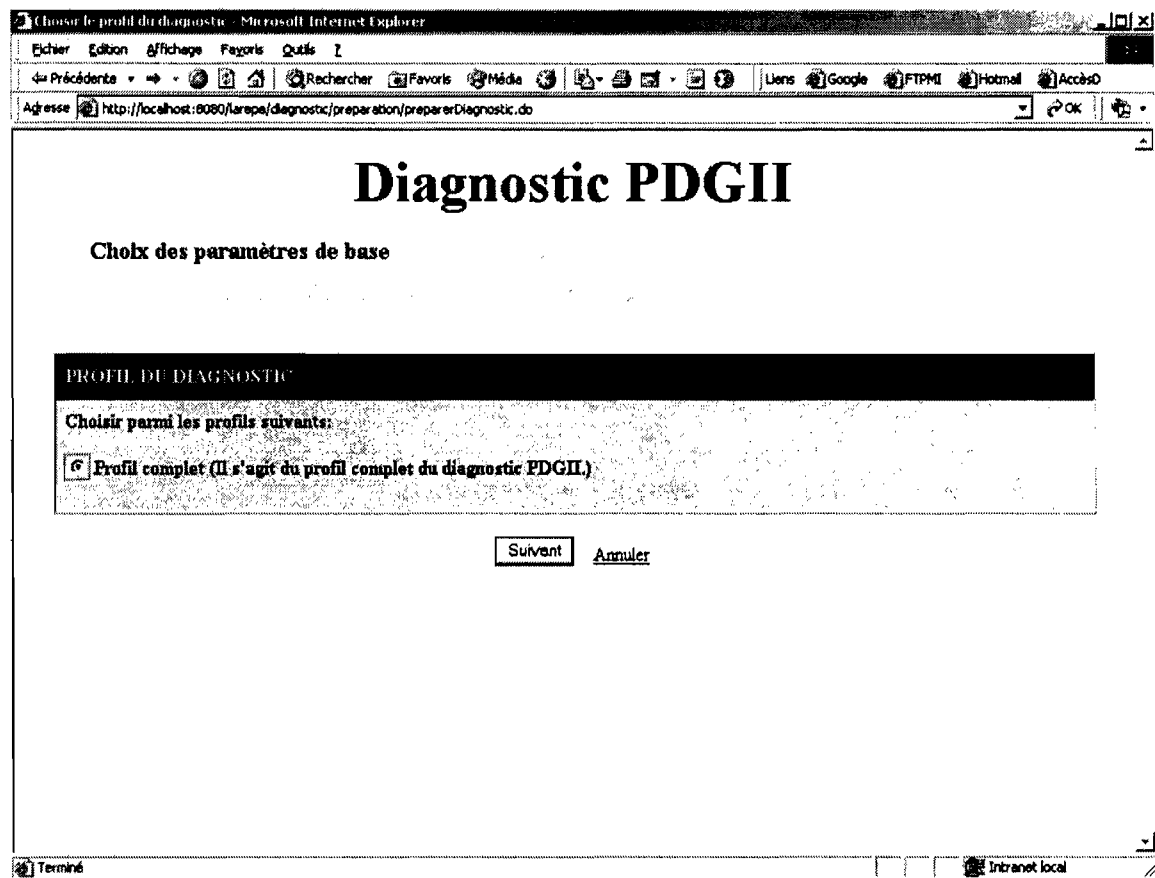


Figure 4.6 Choix du profil de diagnostic

Par la suite, l'utilisateur est rendu à l'étape du choix de l'entreprise sur laquelle portera le diagnostic (figure 4.7).

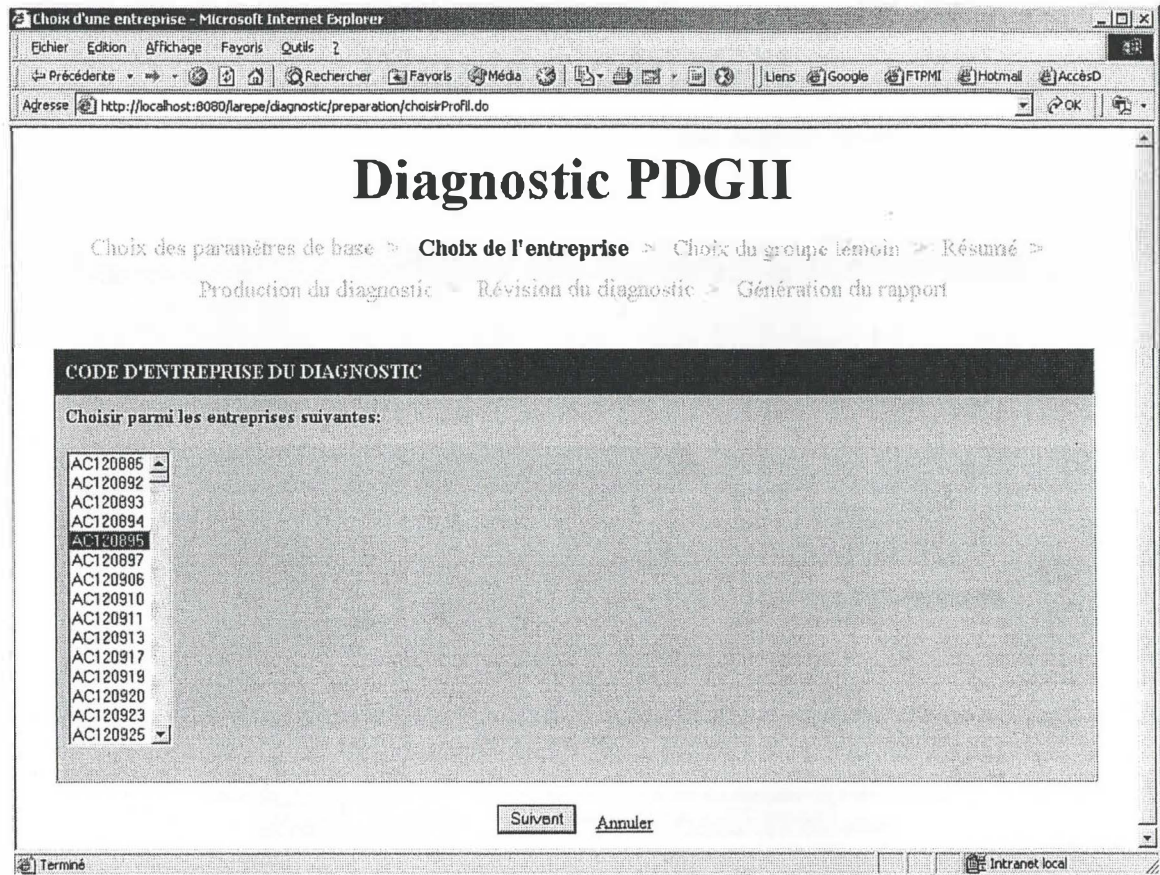


Figure 4.7 Choix de l'entreprise

L'étape suivante consiste au choix du groupe témoin. L'utilisateur doit d'abord choisir la devise dans laquelle se feront les calculs des données de l'entreprise et du groupe témoin. Ensuite, il doit sélectionner les critères de sélection du groupe témoin.

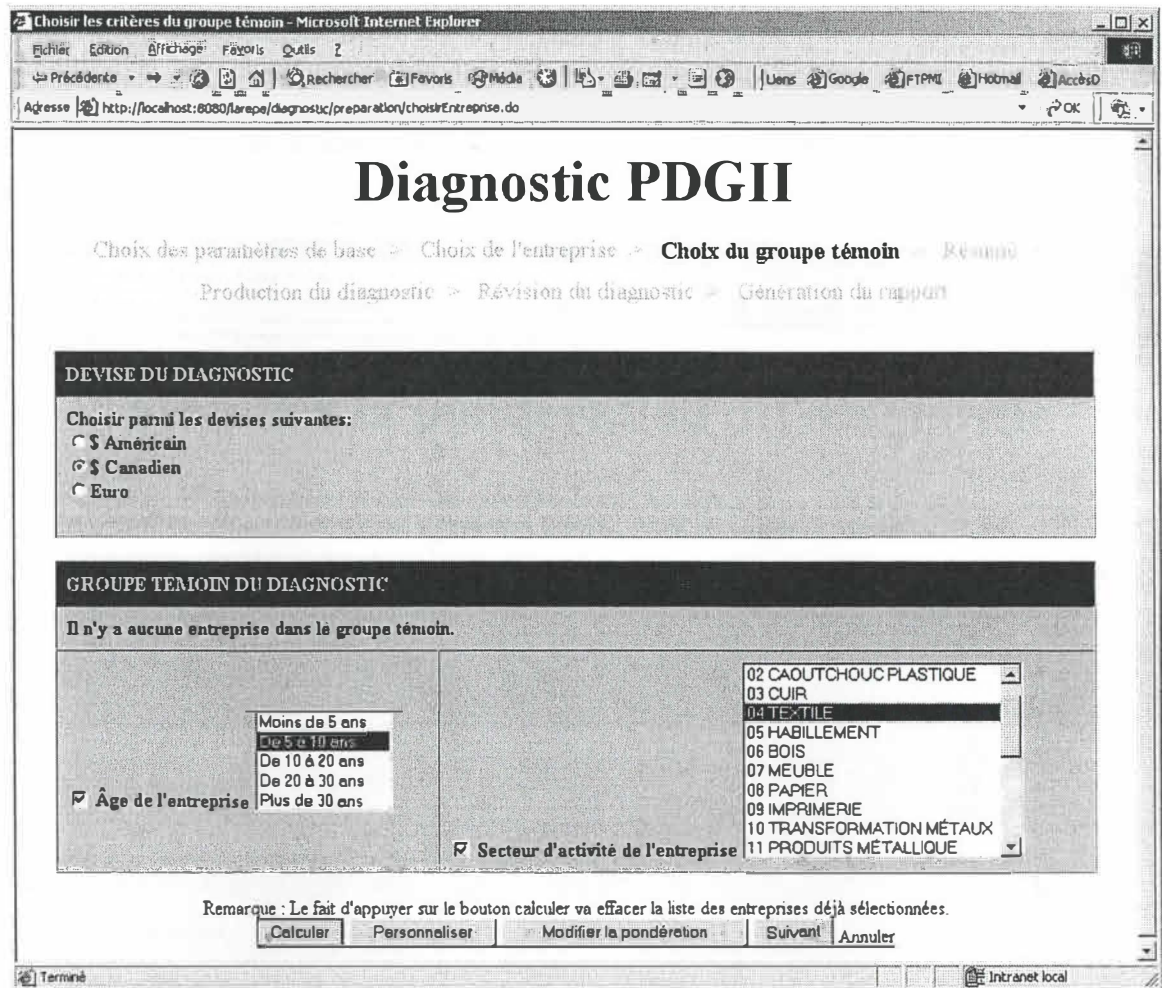


Figure 4.8 Choix de la devise et du groupe témoin

Une étape intermédiaire, accessible à partir de l'étape du choix du groupe témoin (figure 4.8), est celle où il est possible de modifier des éléments d'expertise du diagnostic. Ces modifications n'ont toutefois aucune incidence sur la base de connaissances. Cette manière de fonctionner a des avantages

- Permet à un expert de faire des simulations pour le diagnostic d'une même entreprise avec des valeurs d'expertise différentes, pour ajouter de la précision ou à des fins de recherche.
- Permet à un expert de tester et de valider un éventuel changement d'expertise dans la base de connaissances.

En effet, dans la figure 4.9, l'utilisateur peut voir tous les critères d'évaluation correspondant au profil de diagnostic qu'il a choisi. Ces critères sont bien entendu



regroupés en secteurs d'activités. Pour chacun des critères d'évaluation, la pondération sectorielle par défaut (expertise que l'on retrouve dans la base de connaissances) est présentée. Il est possible pour un utilisateur expert de modifier ces pondérations sectorielles pour les différentes raisons énumérées précédemment.

**Modifier la pondération du diagnostic**

Il est possible de modifier les pondérations de chaque critère à l'aide de cette page. Faire attention afin que le total de chaque section donne bien 100%. Les pondérations par défaut proviennent de la base de connaissance. Cliquez sur "Suivant" pour poursuivre le diagnostic.

Section Systèmes de production	
Conformité des équipements de production	20.00 %
Degré de maîtrise technologique	20.00 %
Investissement dans la production	10.00 %
Normes de qualité	5.00 %
Intégration globale des systèmes de production	10.00 %
Efficacité de la production	20.00 %
Degré d'utilisation des équipements de production	15.00 %

Section Gestion de la production	
Fréquence du contrôle des coûts de la qualité	14.00 %
Fréquence du contrôle des coûts de logistiques	13.00 %
Mesure des éléments d'amélioration	16.50 %

Terminé Intranet local

Figure 4.9 Modification d'expertise : pondération des critères d'évaluation

Par la suite, un résumé des différents paramètres choisis est présenté (figure 4.10). L'utilisateur peut alors valider les paramètres du diagnostic avant sa production. Si tout est correct, la production du diagnostic peut alors être lancée.

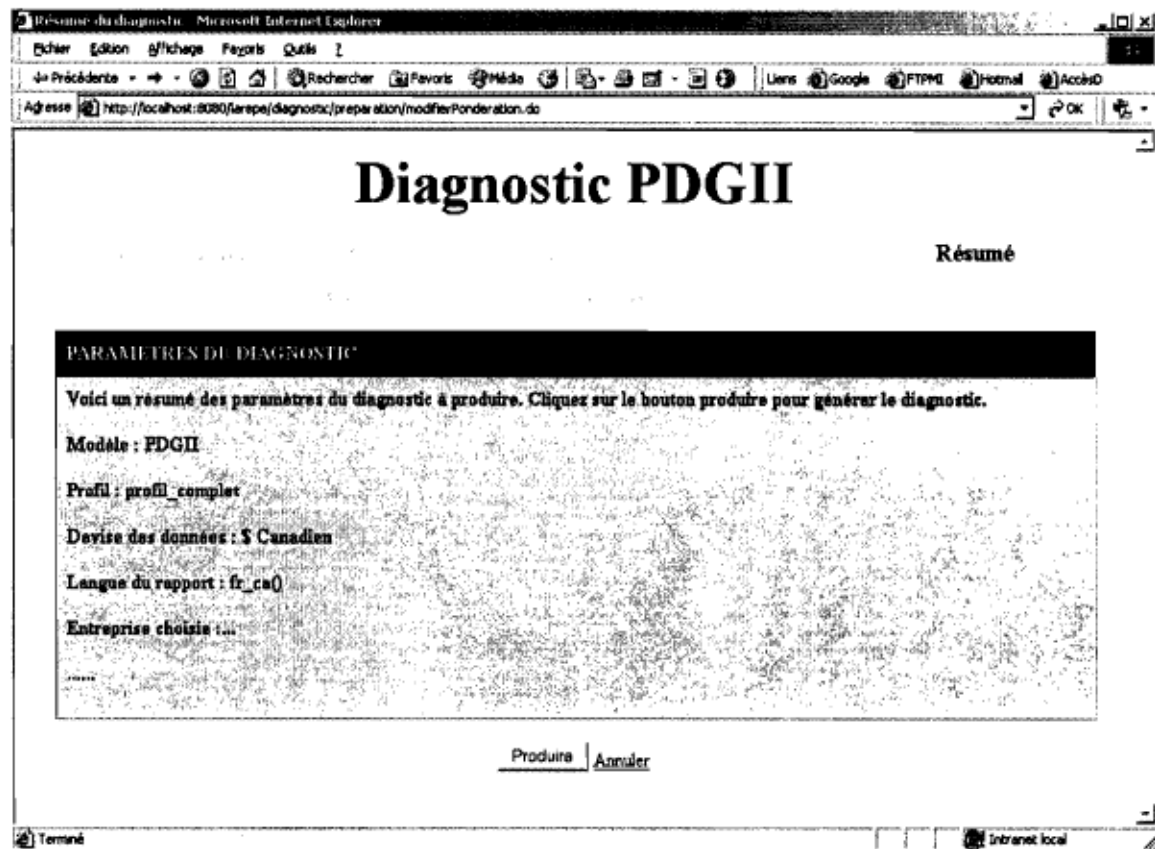


Figure 4.10 Présentation d'un résumé des paramètres du diagnostic

L'étape de production du diagnostic dure en moyenne 1 minute, selon le profil de diagnostic choisi. Durant cette étape (figure 4.11), qui est pleinement automatisée, plusieurs tâches sont effectuées par le système PDGII.

- Tous les paramètres du diagnostic sont sauvegardés dans l'entrepôt de données. Par conséquent, nous conservons une trace de tous paramètres et expertise importante utilisée ainsi que de toutes modifications de l'expertise.
- Il y a extraction de toutes les données nécessaires au diagnostic. Ces données proviennent de l'entrepôt de données.
- Il y a également création des données du groupe témoin (calculs de moyennes, médianes, etc.).
- Le diagnostic même est ensuite exécuté à l'aide de la base de connaissances et du moteur d'inférence.
- Tous les résultats du diagnostic sont sauvegardés dans l'entrepôt de données.

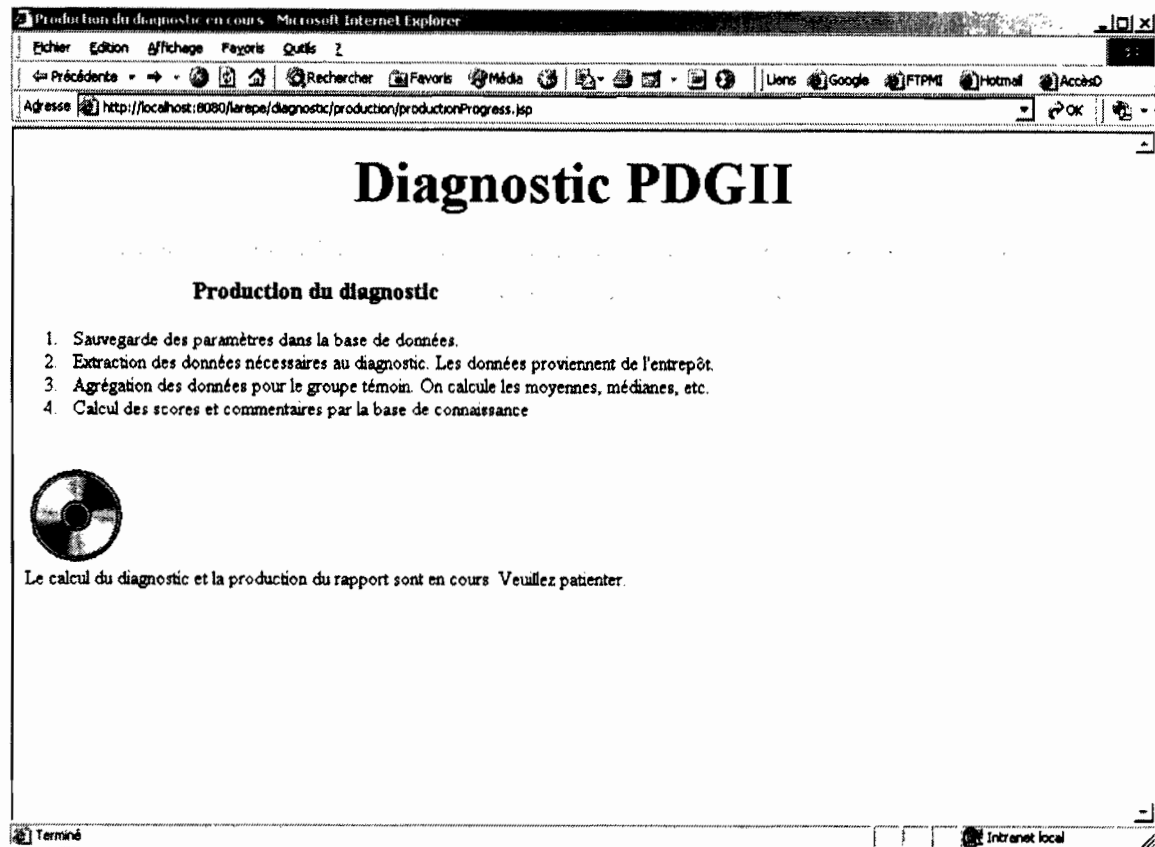


Figure 4.11 Production du diagnostic PDGII par le système

Une fonctionnalité utile et conviviale est la configuration du rapport de diagnostic qu'il est possible de faire une fois le diagnostic terminé. En effet, une fois le diagnostic terminé, tous les éléments du rapport ont été produits (graphiques, commentaires, etc.) et il ne reste qu'à les valider pour qu'un rapport PDF soit produit en quelques secondes.

Cette étape de validation du rapport avant sa production est très simple pour l'utilisateur. Comme on peut le voir à la figure 4.12, l'utilisateur a accès à chacune des sections du rapport et peut en modifier le contenu manuellement. Parmi les modifications qu'il peut faire, on retrouve :

- La modification du titre des graphiques.
- La modification des échelles des graphiques.
- La modification des valeurs minimums et maximums des axes des graphiques.
- La modification des commentaires de recommandation textuels.

Bien entendu, ces modifications sont sauvegardées dans l'entrepôt de données. Une fois que toutes les modifications voulues des éléments du rapport ont été effectuées, la production du rapport en un fichier PDF peut alors être lancée.

Voici dans la figure 4.12, un exemple de l'écran dans lequel on peut faire la modification d'un graphique.

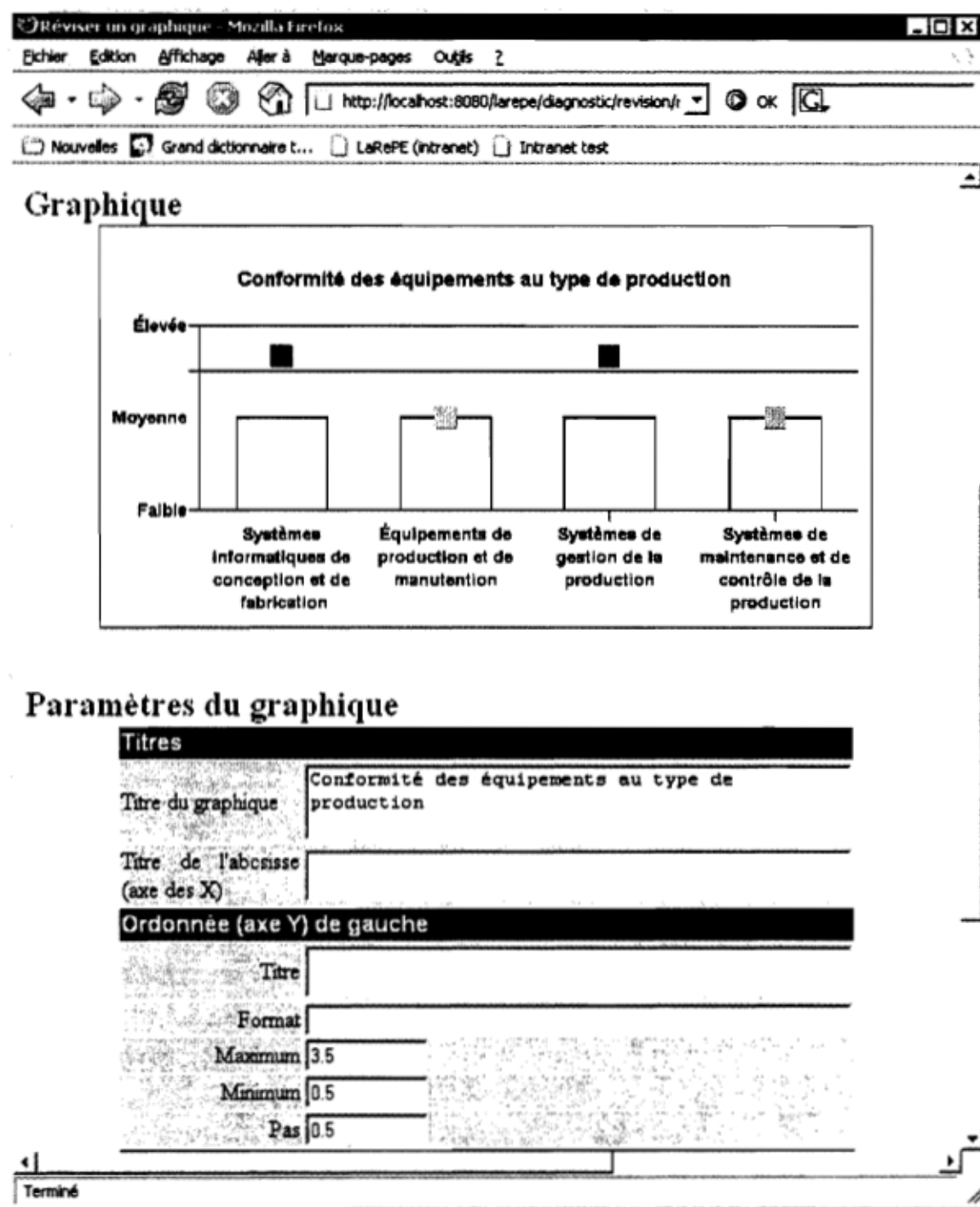


Figure 4.12 Configuration du rapport : modification d'un graphique

Voici maintenant en figure 4.13, un exemple d'écran permettant la modification d'un commentaire de recommandation d'une section.

**Révision d'un commentaire**

**Instructions**

Le commentaire actuel est présenté dans la case de saisie ci-dessous. Il est possible de modifier ce commentaire. Pour conserver la modification, cliquer sur Sauvegarder. Tous les commentaires sont conservés et affichés dans la liste au bas de cette page. Cependant, seul le plus récent commentaire est affiché dans le rapport.

**Commentaire**

Une modification des pratiques actuelles notamment par une intégration informatique des systèmes/technologies contribuerait à réduire les différents délais liés à la production et accroître son efficacité.

**Commentaires précédents**

Personne	Date	Commentaire
KB	2005-09-19 09:25	Une modification des pratiques actuelles notamment par une intégration informatique des systèmes/technologies contribuerait à réduire les différents délais liés à la production et accroître son efficacité.

© Copyright 2003-2004 LaRePE (UQTR). Tous droits réservés

Figure 4.13 Configuration du rapport : modification d'une recommandation




Une autre fonctionnalité intéressante offerte par le programme utilisateur est celle qui permet l'accès à un diagnostic déjà produit. Cette fonctionnalité amène plusieurs avantages dont celui qui permet aux experts de pouvoir refaire le même diagnostic, mais avec des éléments d'expertise différents, par exemple les pondérations des critères qui sont configurables lors d'un diagnostic. Il est donc possible d'obtenir plusieurs essais pour un même diagnostic et tous les résultats de chaque essai, allant des options de configuration, passant par les valeurs des paramètres d'expertise configurables, jusqu'aux résultats de l'essai du diagnostic sont sauvegardés dans l'entrepôt de données.

En somme il est possible, à partir de la liste de diagnostics déjà produits, de :

- Consulter un diagnostic.
- Imprimer un diagnostic.
- Réviser et reproduire un diagnostic.

Voici en figure 4.14, un exemple de l'écran dans lequel on retrouve la liste des diagnostics déjà produits.


**Diagnostic**
Accueil  
Déconnexion

## Liste des diagnostics #

### Instructions

Voici la liste des diagnostics déjà faits pour le modèle #.

- Pour créer un nouveau diagnostic (rapport), cliquer sur "Créer un nouveau diagnostic".
- Pour consulter, réviser ou imprimer un rapport, cliquer sur le code d'entreprise correspondant.

**Opérations sur les diagnostics**  
[Créer un nouveau diagnostic](#)

Code d'entreprise	Date de production	Date de transmission
<u>DEMO0000</u>	25-08-2005 (14:31)	
<u>MC120677</u>	25-08-2005 (14:14)	
<u>PP000103</u>	25-08-2005 (11:33)	
<u>PP000122</u>	25-08-2005 (08:42)	
<u>PP000122</u>	24-08-2005 (14:45)	
<u>GR121010</u>	24-08-2005 (14:39)	
<u>PP000122</u>	24-08-2005 (14:34)	
<u>GR121010</u>	23-08-2005 (16:16)	
<u>PP000052</u>	23-08-2005 (08:21)	

Terminé

Figure 4.14 Liste des diagnostics déjà produits d'un utilisateur

#### **4.2.2.2 Solutions apportées**

En résumé, le programme utilisateur possède de nombreuses qualités que son prédécesseur du système PDG ne possédait pas totalement.

- Il permet la configuration des paramètres de base d'un diagnostic.
- Il permet la configuration de certains éléments d'expertise pour le diagnostic.
- Il permet la configuration du rapport de diagnostic avant sa production en PDF (graphiques, commentaires de recommandation).
- Il permet la reproduction d'un diagnostic qui a déjà été produit. Cela peut être fait grâce à l'entrepôt de données (voir section suivante).

#### **4.2.3 Entrepôt de données**

L'entrepôt de données du LaRePE est venu faciliter grandement le développement du système PDGII. Comme nous le verrons dans cette section, cet entrepôt de données a résolu quelques limites que possédait le système PDG.

##### **4.2.3.1 Conception et technologie**

Tout d'abord, il est important de spécifier que l'entrepôt de données du LaRePE n'a pas été développé durant ce projet de maîtrise. Il a été développé par Mathieu Dugré dans le cadre de son propre projet de maîtrise durant les années 2003 et 2004 [4, 7].

Comme il ne s'agit pas du sujet de ce projet de maîtrise, nous n'entrerons pas dans les détails. Nous pourrions seulement dire qu'un entrepôt de données peut regrouper les données de plusieurs bases de données afin de les réorganiser afin qu'elles soient disponibles à des fins d'analyses et d'utilisations dans des systèmes comme le PDGII. Pour des détails plus techniques, voir [4, 7].

Le système de gestion de bases de données Oracle a été utilisé pour le développement de l'entrepôt du LaRePE.

##### **4.2.3.2 Solutions apportées**

L'entrepôt de données [4, 7] est venu résoudre plusieurs limites du système PDG. Il a entre autres permis l'abandon du logiciel SAS qui était utilisé pour des calculs statistiques pour le diagnostic. Le programme SAS de calculs statistiques devait être utilisé chaque fois qu'un diagnostic devait être produit. Grâce à l'entrepôt, une grande partie des calculs statistiques sont faits lors d'une routine que l'entrepôt exécute

quotidiennement durant la nuit. Pour les autres calculs qu'il reste à produire à chaque diagnostic, comme par exemple les calculs pour la création des données du groupe témoin, ils sont produits rapidement par l'entrepôt au moment du diagnostic. Le temps d'exécution d'un diagnostic PDGII est donc amélioré grâce aux calculs en moins à faire (au moment du diagnostic) ainsi qu'à l'utilisation de l'entrepôt pour la création des données du groupe témoin.

L'entrepôt de données permet également la sauvegarde des données de chaque diagnostic produit par le système. Chaque fois qu'un diagnostic est produit, les données suivantes sont sauvegardées dans l'entrepôt :

- Les données de l'entreprise et du groupe témoin qui ont servi au diagnostic.
- Les résultats détaillés du diagnostic.
- Les valeurs des éléments d'expertise utilisés (pondération des critères d'évaluation, etc.).
- Les modifications apportées au rapport (graphiques, commentaires, etc.).

La sauvegarde de toutes ces informations de chaque diagnostic permet de reproduire n'importe quel diagnostic avec les mêmes données et la même expertise. Cela permet aussi à un chercheur de faire plusieurs simulations de diagnostic d'une entreprise et de comparer les résultats (voir section 4.2.2 sur le programme utilisateur).

#### **4.2.4 Le rapport de diagnostic**

Le rapport de diagnostic est un élément clé dans le système PDGII. C'est dans ce rapport que l'entrepreneur retrouve en détail tous les résultats concernant la performance de son entreprise. Comme nous le verrons dans cette section, la principale qualité du rapport du système PDGII est son caractère générique qui, grâce à sa programmation, permet facilement la configuration de différentes mises en page. Il est important de mentionner que comme le système PDGII n'est pas encore en utilisation, la mise en page de son rapport n'a pas été finalisée. Tout ce qui traite de mise en page dans cette section est sujet à changement dans le futur. Cependant, tout ce qui concerne la programmation et le caractère générique de ce nouveau rapport est final.

##### **4.2.4.1 Présentation du rapport**

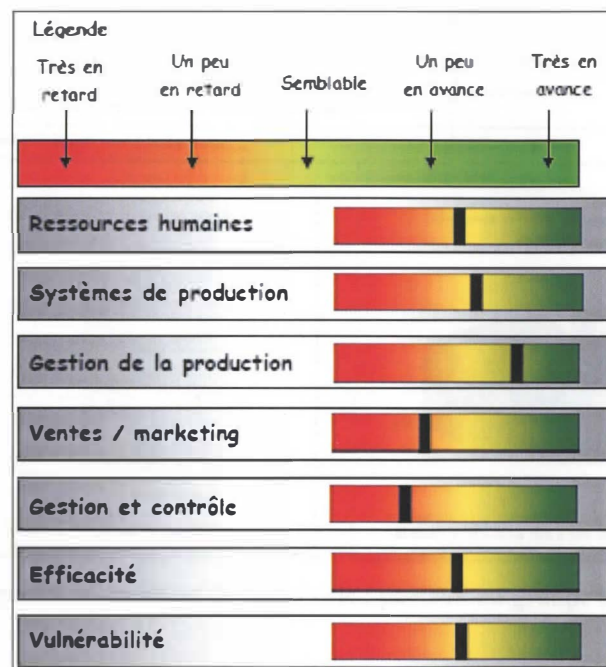
Le rapport contient 4 parties :

- La page résumé de la performance de l'entreprise.
- Les pages d'information sur le groupe témoin.
- Les pages d'évaluation des différents secteurs d'activités de l'entreprise.

- La page du plan d'action.

#### La page résumé de la performance de l'entreprise

Il s'agit de la page couverture du rapport de diagnostic. Cette page permet à l'entrepreneur de visualiser rapidement le niveau performance de son entreprise sur chacun des secteurs d'activités qui ont été évalués. Ces secteurs sont définis selon le profil de diagnostic choisi (voir chapitre 4.2.1.1). Afin de permettre cette visualisation rapide, un tableau résumant la performance de l'entreprise se retrouve sur cette première page comme en témoigne la figure 4.15.



**Figure 4.15** Rapport de diagnostic : portrait global de la performance de l'entreprise

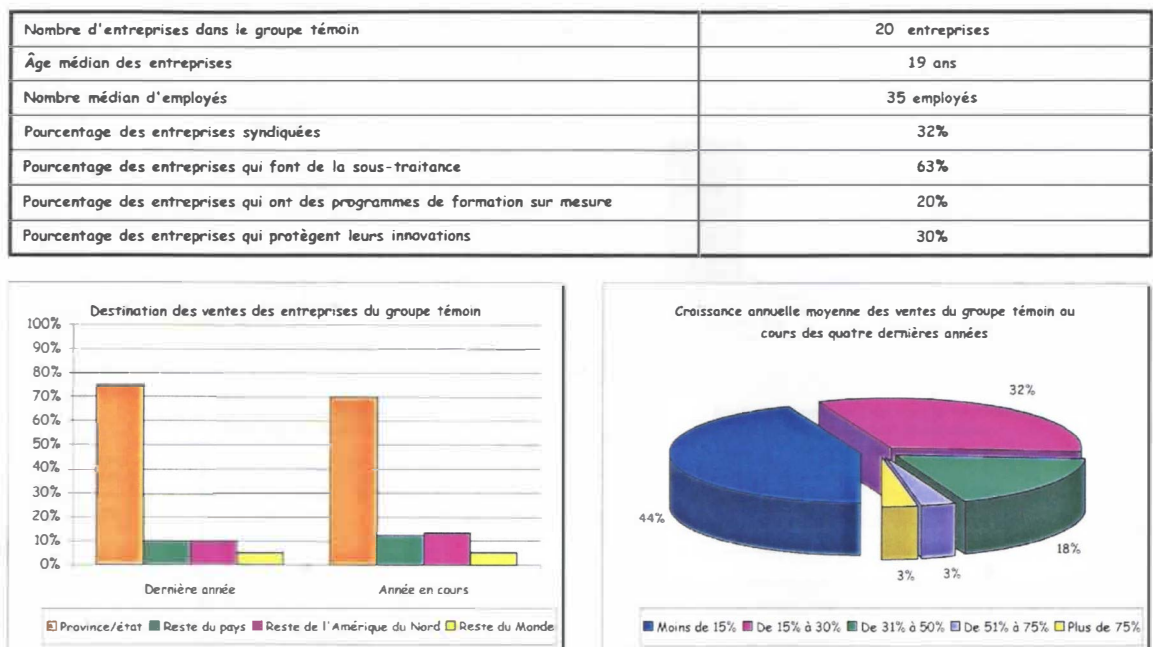
Le niveau de performance est présenté sur une échelle qualitative, situant l'entreprise par rapport à son groupe témoin, derrière laquelle se trouvent les critères et données quantitatives.

#### Les pages d'informations sur le groupe témoin

Il s'agit de pages présentant en détail le groupe témoin. Ces informations sur le groupe témoin sont présentées sous forme de tableaux et de graphiques.

L'information la plus pertinente pour le dirigeant de l'entreprise consiste dans la présentation des critères de sélection qui ont permis la création du groupe témoin. Il s'agit d'une information pertinente pour le dirigeant puisque qu'elle lui indique avec quels types d'entreprises son entreprise a été comparée.

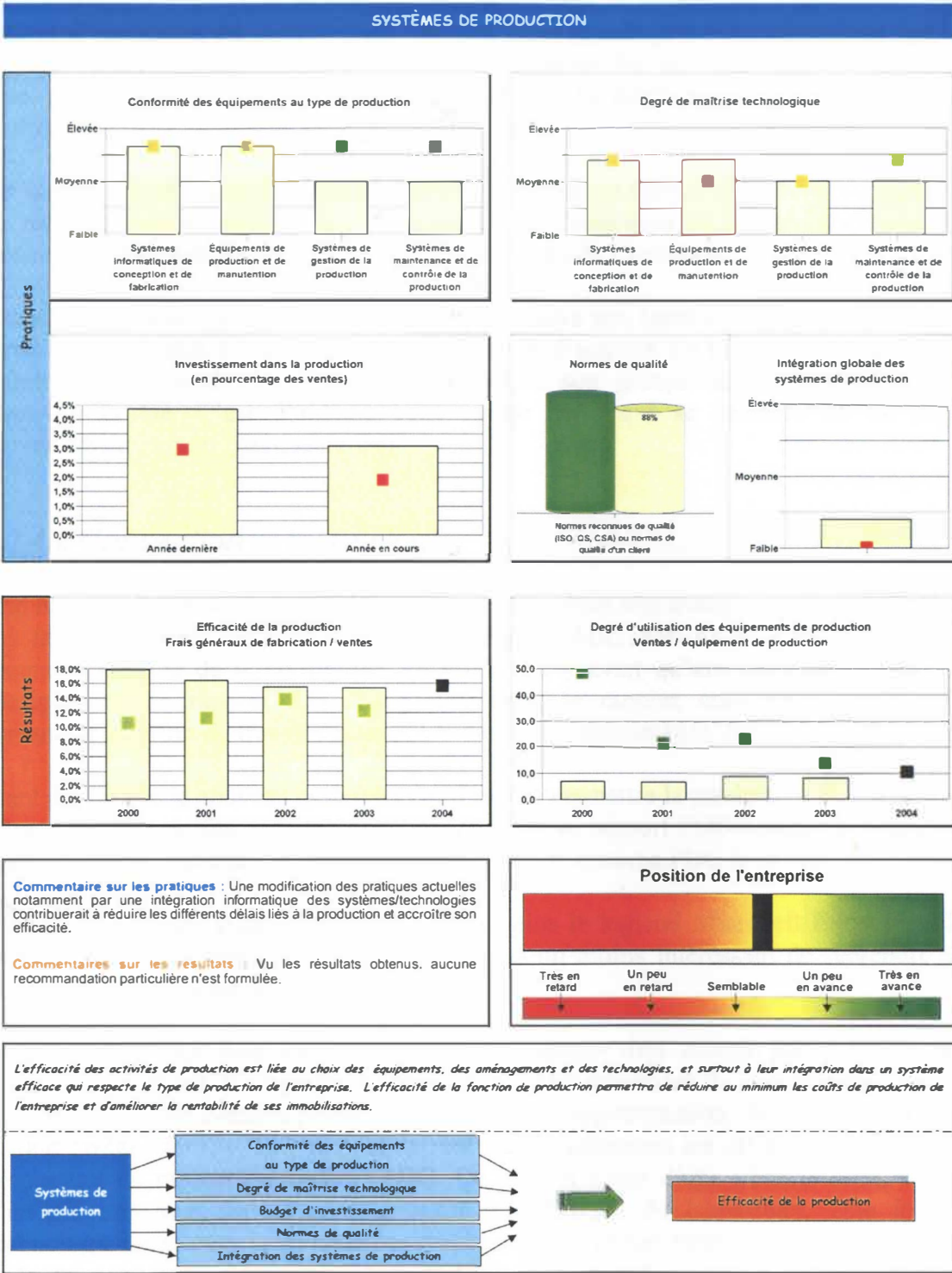
On retrouve aussi plusieurs informations qui n'ont pas été utilisées pour définir le groupe témoin, mais qui aident à indiquer à l'entrepreneur de quelle manière évoluent les entreprises en faisant partie. En voici quelques exemples (figure 4.16).



**Figure 4.16 Rapport de diagnostic : portrait du groupe témoin**

### Les pages d'évaluation des différents secteurs d'activités de l'entreprise

Dans cette partie du rapport, on retrouve les résultats détaillés de la performance de l'entreprise. On compte une page par secteur d'activité évalué. Ces secteurs d'activités sont déterminés selon le profil choisi pour le diagnostic. Chacune de ces pages est construite avec la même logique (sauf pour les pages concernant l'efficacité et la vulnérabilité qui n'évaluent respectivement que des résultats et des pratiques, et non les deux). La figure 4.17 montre un exemple d'une de ces pages. Dans ce cas-ci, il s'agit de la page des résultats de la performance du secteur d'activité des systèmes de production.



Tout d'abord, on retrouve des graphiques présentant la performance de chacun des critères d'évaluation du secteur d'activité. Ces critères sont présentés selon deux types. Il y a premièrement les critères portant sur l'utilisation des pratiques de l'entreprise et juste en-dessous, il y a les critères portant sur la performance des résultats de l'entreprise.

Par la suite, on retrouve les commentaires sous forme de textes. Il y a un commentaire de recommandation portant sur l'évaluation des pratiques et un commentaire de recommandation portant sur l'évaluation des résultats de l'entreprise.

Finalement, à la droite des commentaires, on retrouve une barre synoptique illustrant la performance globale de l'entreprise pour le secteur d'activité. Ce niveau de performance définit la position de l'entreprise par rapport à son groupe témoin sur le secteur d'activité évalué. C'est d'ailleurs cette même barre qui se retrouve sur la page couverture du rapport (page résumée).

#### **4.2.4.2 Conception et technologie**

Une limite importante du rapport du système PDG, était son manque de flexibilité dû à aux faits qu'il avait été développé avec le logiciel Microsoft Excel et qu'il n'y avait aucune possibilité de le configurer. En effet, il n'y avait qu'une seule possibilité de configuration de rapport. Un changement dans le rapport était très complexe et demandait un effort de travail considérable dans la programmation Excel.

Étant donné que le nouveau système PDGII devait permettre la production de diagnostic selon différents profils, il fallait donc que le nouveau rapport PDGII permette plusieurs configurations différentes et non une seule comme le système PDG le permettait.

Comme le nouveau système PDGII n'utilisait plus le logiciel Microsoft Excel pour la programmation du diagnostic, il aurait été plus ou moins intéressant de reprendre ce logiciel pour la programmation du rapport.

Nous nous sommes donc tournés vers une technologie déjà utilisée par le laboratoire pour un autre projet. Il s'agit de la librairie NetChart provenant de la compagnie Visual Mining [32]. Cette librairie permet la création, par programmation, de graphiques d'une qualité professionnelle. Elle permet de reproduire facilement les différents modèles de graphiques du rapport du système PDG. De plus, il s'agit d'une librairie Java, ce qui s'intègre très bien dans l'environnement informatique du laboratoire. Une fois la technologie permettant la création des graphiques du rapport trouvée, il restait à trouver celle qui permettrait de produire le rapport lui-même et d'y insérer les graphiques créés avec la librairie NetChart.



Nous avons eu l'idée qu'il serait très utile d'obtenir les rapports dans un format PDF. Ce format était déjà largement utilisé par le laboratoire et il est facile d'impression. Une recherche de différents outils permettant la programmation de fichiers PDF a donc été menée. Nous avons trouvé une technologie très intéressante qui se nomme PDFLIB [21]. Grâce à cette technologie, il est possible de rajouter des « blocks » à un fichier PDF existant. Un « block » est une zone ayant un emplacement et une grandeur définis permettant l'ajout par programmation de textes ou d'images.

La figure 4.18 représente une page modèle du rapport PDGII pour la section des ressources humaines. Comme on peut le voir, nous y avons placé des « blocks » qui pourront recevoir par programmation du texte ou des graphiques (créés par notre librairie NetChart) lors de la création du rapport.

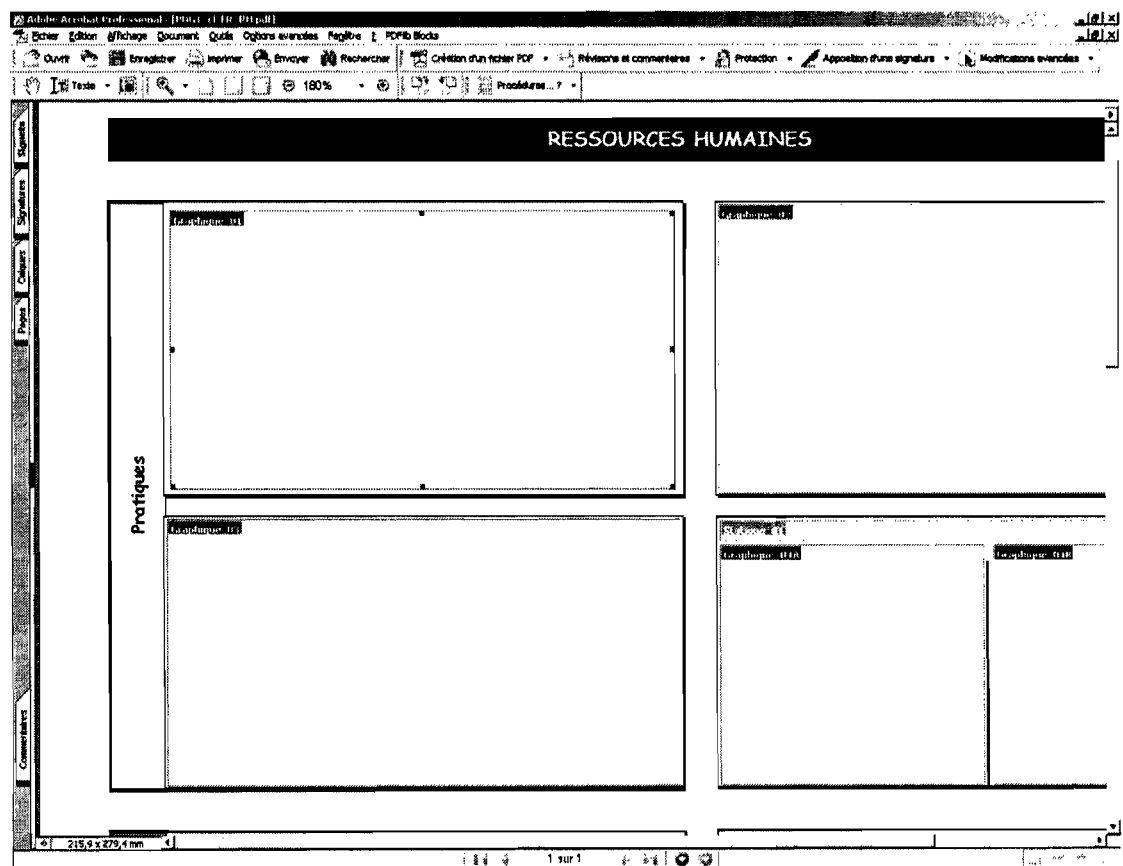


Figure 4.18 Exemple d'un « block » dans un PDF

De plus, chaque « block » peut être configuré par programmation, ou à l'aide de l'interface PDFLIB dans Acrobat comme on peut le voir dans la figure 4.19.

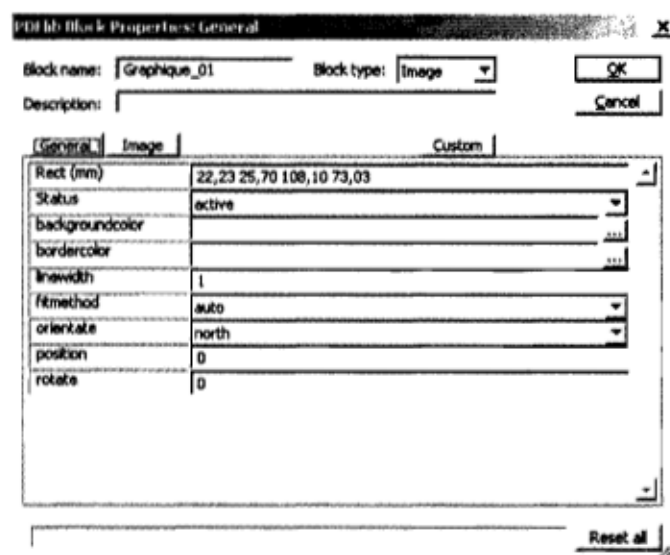


Figure 4.19 Configuration d'un « block »

Nous avons donc créé des modèles de fichiers PDF pour chaque page du rapport à l'aide du logiciel Adobe Acrobat (<http://www.adobe.com>). Donc, chaque fois qu'un nouveau secteur d'un profil est créé dans la base de connaissances pour le diagnostic, il faut seulement faire la création de la page modèle PDF de ce secteur à l'aide du logiciel Acrobat et créer les « blocks » avec la librairie PDFLIB dans la page modèle PDF aux bons endroits.

#### 4.2.4.3 Solutions apportées

Le système PDGII utilise donc les technologies suivantes :

- Le logiciel Acrobat pour la création des différentes pages modèles de différents profils. Ce logiciel est utilisé seulement lorsqu'il faut faire la création d'une nouvelle page modèle. Une fois la page créée, elle est utilisable pour n'importe quel nombre de diagnostics.
- La librairie PDFLIB pour l'ajout des différentes zones (« blocks ») de graphiques et de textes dans les pages modèles de différents profils. L'ajout de zones (« blocks ») se fait lorsqu'il y a création d'une nouvelle page modèle. Cette opération est faite une seule fois, lors de la création de la page modèle.

- La librairie NetChart pour la création des graphiques qui seront insérés dans les différentes zones (« blocks ») des pages modèles nécessaires au profil du diagnostic choisi. Cette librairie est utilisée chaque fois qu'un rapport de diagnostic doit être produit.

Grâce à ces technologies, le rapport du système PDGII a acquis les qualités suivantes :

- Il est flexible et configurable selon le profil de diagnostic que l'utilisateur désire.
- Il est très simple et rapide de créer les pages modèles de nouveaux profils éventuels.
- Il est possible de modifier le contenu du rapport avant sa création définitive (voir chapitre 4.2.2).

Le prochain chapitre portera sur les tests qui ont été effectués afin de valider la base de connaissances et les résultats produits par le système PDGII. De plus, une discussion concernant de futurs tests pour d'éventuelles mises à jour de la base de connaissances sera faite. Ensuite, une méthode d'évaluation des résultats de diagnostic du système PDGII sera proposée. Finalement, un exemple complet de toutes les étapes du processus d'un diagnostic du système PDGII sera présenté.

## **CHAPITRE 5**

### **TESTS, VALIDATION, EXEMPLE**

Tout système expert se doit d'être testé en profondeur afin d'en valider les résultats. La partie la plus importante et critique à tester d'un système expert est sans contredit la base de connaissances, cette dernière étant le noyau central du fonctionnement du système expert. Ce chapitre portera sur les tests qui ont été effectués pour valider le contenu de la base de connaissances ainsi que les résultats produits par le système PDGII. De plus, nous suggérerons une méthode pour les futurs tests des éventuelles mises à jour de la base de connaissances. Finalement, un exemple complet contenant toutes les étapes de production d'un diagnostic à l'aide du système PDGII sera présenté.

#### **5.1 Tests et validation**

Il était très important d'exécuter différents tests afin de valider les éléments d'expertise de la base de connaissances. Nous avons basé nos tests sur deux principaux faits.

1. Les éléments d'expertise du système PDG sont valides et testés depuis plusieurs années.
2. Les éléments d'expertise se retrouvant dans la base de connaissances du système PDGII correspondent aux mêmes éléments d'expertise qui se retrouvent dans l'implémentation Excel du système PDG (mais avec une toute autre représentation et structure).

Par conséquent, afin de tester la base de connaissances du système PDGII, nous avons exécuté une série de diagnostics avec les deux systèmes, le PDG et le PDGII, et ce, à l'aide des mêmes données. Ensuite, pour chaque exécution, nous avons fait la comparaison des résultats détaillés pour chaque étape de raisonnement du diagnostic, à la recherche de différences dans les résultats du système PDGII par rapport à ceux du système PDG, ce dernier contenant les bons résultats validés depuis plusieurs années.

Plus concrètement, des jeux de données de 5 diagnostics ont été produits. Les tests de ces 5 diagnostics furent faits de manière itérative. C'est-à-dire que chaque fois qu'un test de diagnostic relevait une différence dans les résultats du système PDGII par rapport à ceux du système PDG, et qu'une modification dans la base de connaissances était nécessaire, tous les tests précédents devaient être refaits avec la base de connaissances corrigée, peu importe la nature de cette modification, aussi petite soit-elle.

Cette méthode itérative de test est illustrée sur la figure 5.1.

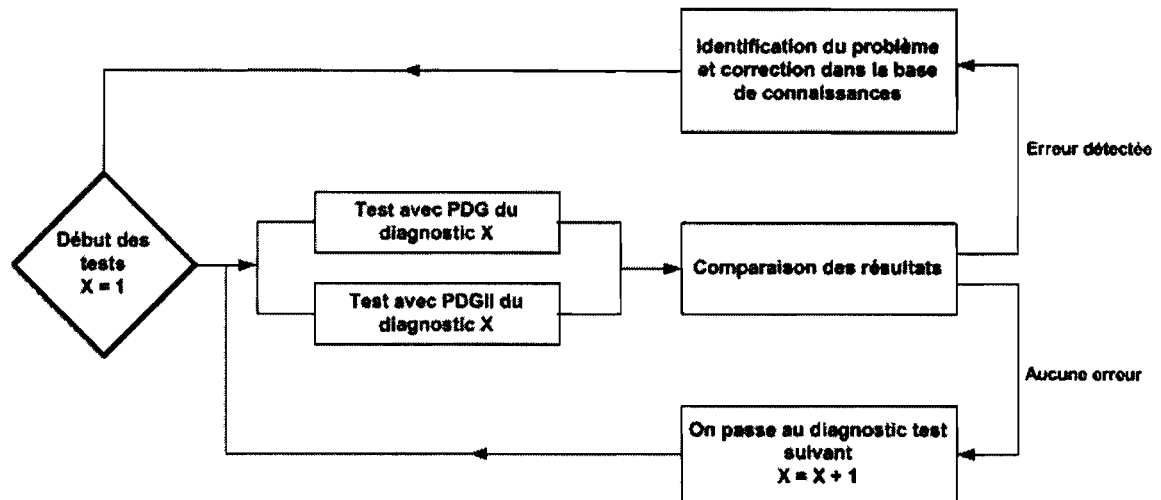


Figure 5.1 Processus itératif des tests

Voyons maintenant un exemple parmi ces 5 diagnostics tests qui ont été exécutés. Nous verrons un diagnostic test ne contenant aucune erreur, c'est-à-dire aucune différence entre les résultats de ce diagnostic dans le système PDG et le nouveau système PDGII.

Pour bien comprendre les tests, rappelons-nous qu'un diagnostic est constitué de plusieurs critères d'évaluation. Ces critères d'évaluation sont regroupés en différents secteurs d'évaluation. La **première étape** d'un test consiste donc à vérifier le résultat (le score) de chacun des critères d'évaluation.

La **deuxième étape** consiste à vérifier les résultats au niveau de chaque secteur d'évaluation. Ce qui ressort de l'évaluation globale de chaque secteur d'évaluation, est l'identification des critères d'évaluation qui représentent des faiblesses pour l'entreprise dans chaque secteur. Le nombre maximal de critères d'évaluation que le système PDGII identifie est réglé par un paramètre qui est présentement par défaut à 1. Ce qui veut dire que lors des tests, le système PDGII (ainsi que le PDG) a identifié, s'il y a lieu, un critère d'évaluation faible concernant les pratiques et un autre concernant les résultats de l'entreprise, et ce, pour chaque secteur d'évaluation.

La **troisième étape** concerne encore l'évaluation globale de chaque secteur d'évaluation, mais exécute la vérification des recommandations produites par le système pour chaque secteur d'évaluation. Dans cette étape, il suffit seulement de vérifier si les recommandations produites sont les mêmes pour le système PDG et le système PDGII.

Le tableau 5.1 présente les résultats de ces 3 étapes, sur un test qui ne comporte aucun problème.

ÉTAPE 1					ÉTAPE 2	
Secteur d'évaluation	Critères d'évaluation	Score PDGII	Score PDGI	Comparaison score	Critères faibles PDGII	Critères faibles PDGI
Systèmes production	conformite equipements production	3	3	OK		
	degre maitrise technologique	2	2	OK		
	investissement production	1	1	OK		
	Normes qualite	-200	-200	OK		
	integration globale systemes production	0	0	OK	1	1
	efficacite production	4	4	OK		
	degre utilisation equipements production	3	3	OK		
Gestion production	frequence controle couts qualite	2.67	2.67	OK		
	frequence controle couts logistiques	2.8	2.8	OK		
	mesure elements amelioration	2	2	OK		
	pourcentage ventes RD equipements	4	4	OK		
	pourcentage ventes RD procedes	4	4	OK		
	age moyen stocks 5ans	3	3	OK		
	efficacite gestion production	3.42	3.42	OK		
	jointure pourcentage ventes RD	4	4	OK		
Ressources humaines	Formation	2	2	OK		
	gestion participative	1	1	OK	1	1
	niveau diffusion informations	2.67	2.67	OK		
	remuneration axee performance personnel administratif	3	3	OK		
	remuneration axee performance personnel production	3	3	OK		
	evaluation rendement personnel administratif	3	3	OK		
	evaluation rendement personnel production	3	3	OK		
	taux depart volontaires cadres	2	2	OK		
	taux depart volontaires personnel production	1	1	OK		
	qualite relations travail	2	2	OK		

	productivite_personnel_administratif	1	1	OK		
	productivite_personnel_production	1.5	1.5	OK		
	jointure_remuneration_axe_performance	3	3	OK		
	jointure_evaluation_rendement	3	3	OK		
	jointure_taux_departs_volontaires	-200	-200	OK		
	jointure_productivite	1.25	1.25	OK	1	1
Ventes marketing	pourcentage_ventes_RD_produits	0	0	OK	1	1
	recherche_infos_commerciales	2.5	2.5	OK		
	collaboration_affaires	0.33	0.33	OK		
	effort_vente					
	importance_objectifs_amelioration_continue	3	3	OK		
	suivi_satisfaction_clientele	2	2	OK		
	traitement_plaintes	2	2	OK		
	pourcentage_ventes_produits_nouveauxETmodifies	0	0	OK	1	1
	indice_croissance_ventes	2.25	2.5	Score non identique, mais dû à une erreur de précision du PDGI.		
	jointure_suivi_satisfaction_traitement_plainte	-200	-200	OK		
Gestion et contrôle	fonds_roulement	2	2	OK		
	budget_caisse	0	0	OK		
	systeme_calcul_prix_revient	0	0	OK	1	1
	etats_financiers_previsionnels	2	3	Score non identique, mais dû à une erreur de précision du PDGI.		
	fonctions_avec_responsable_designe	2.25	2.25	OK		
	outils_gestionETplanification	2	2	OK		
	outils_communication	2	2	OK		
	age_moyen_comptes_clients	2	2	OK		
	age_moyen_stocks_3ans	2.24	2.24	OK		
	age_moyen_comptes_fournisseurs	3.2533	3.2533	OK		
	efficacite_gestion_administrative	0	0	OK		
	marge_securite_3ans	0	0	OK	1	1
	jointure_principaux_outils_gestion	-200	-200	OK		

	jointure_niveau_informatisation	-200	-200	OK		
	jointure_age_comptesETstocks	-200	-200	OK		
Efficacité	efficacite_capital_humain	0.5	0.5	OK	1	1
	degre_utilisation_equipements_production	4	4	OK		
	cycle_conversion_encaisse	2.9867	2.9867	OK		
	ratio_fonds_roulement	1.76	1.76	OK		
	marge_brute	2.24	2.24	OK		
	marge_nette	1.32	1.32	OK		
	taux_rendement_fonds_propres	2.24	2.24	OK		
	jointure_marges PROFITS	-200	-200	OK		
Vulnérabilité	releve_direction_assuree	3	3	OK		
	ca_ou_comite_gestion	3	3	OK		
	qualite_relations_travail	2	2	OK		
	degre_dependance_employes_cles	4	4	OK		
	taux_dependance_avec_principaux_clients	3	3	OK		
	taux_dependance_avec_principaux_fournisseurs	2	2	OK		
	valeur_amortie_equipements_production	3	3	OK		
	taux_utilisation_garanties	3.24	3.24	OK		
	marge_securite_5ans	0	0	OK	1	1
	jointure_direction_entreprise	3	-200	Score non identique mais dû à des règles de jointure différentes.		
	jointure_taux_dependance_commerciale	3	-200	Score non identique mais dû à des règles de jointure différentes.		
ÉTAPE 3						
Secteur d'évaluation	Recommandations produites dans le système PDGII					Identique dans le PDGI ?
Systèmes production	Pratique: Une modification des pratiques actuelles notamment par une intégration informatique des systèmes/technologies contribuerait à réduire les différents délais liés à la production et accroître son efficacité.					OUI



	<b>Résultat:</b> Vu les résultats obtenus, aucune recommandation particulière n'est formulée.	OUI
Gestion production	<b>Pratique:</b> À court terme, aucune modification particulière des pratiques utilisées n'est suggérée à l'entreprise.	OUI
	<b>Résultat:</b> Vu les résultats obtenus, aucune recommandation particulière n'est formulée.	OUI
Ressources humaines	<b>Pratique:</b> Une amélioration de la performance de la gestion des ressources humaines pourrait être envisagée, notamment par l'implantation de la gestion participative pour accroître l'implication des employés dans le développement de l'organisation.	OUI
	<b>Résultat:</b> Il peut être nécessaire de porter une attention particulière à la productivité de l'ensemble du personnel afin d'accroître l'efficacité de l'entreprise.	OUI
Ventes marketing	<b>Pratique:</b> Une amélioration de la performance de cette fonction serait possible entre autre par une intensification des activités de R-D concernant les produits afin d'accroître ou maintenir le degré d'innovation de l'entreprise.	OUI
	<b>Résultat:</b> D'après ces résultats, une attention particulière pourrait être portée au degré d'innovation de l'entreprise sur ses produits pour préserver/accroître ses parts de marché et/ou aider à diversifier sa clientèle.	OUI
Gestion et contrôle	<b>Pratique:</b> Une amélioration de la gestion et du contrôle pourrait être envisagée grâce à l'utilisation d'un système de calcul de prix de revient pour connaître les coûts réels de production de l'entreprise et maximiser sa rentabilité.	OUI
	<b>Résultat:</b> Il serait intéressant de porter une attention particulière au niveau de la marge de sécurité afin que l'entreprise dispose d'un coussin de liquidités nécessaires à son développement.	OUI
Efficacité	<b>Résultat:</b> Pour accroître l'efficacité dans l'utilisation des ressources, il serait intéressant d'évaluer la productivité des diverses catégories d'employés afin de réduire la masse salariale et d'augmenter la rentabilité de l'entreprise.	OUI
Vulnérabilité	<b>Pratique:</b> Une façon de réduire la vulnérabilité de l'entreprise serait d'analyser les liquidités et la gestion de celles-ci afin de fournir une plus grande marge de sécurité à l'organisation, nécessaire pour assurer un développement efficace.	OUI

**Tableau 5.1 Test de comparaison entre les résultats du système PDG et ceux du nouveau système PDGII.**

On peut constater dans les résultats de ce test que tous les scores de tous les critères d'évaluation sont identiques pour le système PDG (colonne 4) ainsi que pour le système PDGII (colonne 3). Dans seulement 4 critères d'évaluation, le score n'est pas identique, mais une raison explique cette différence comme spécifié dans le tableau 5.1 des résultats. Une de ces différences est due au fait que la précision des données n'est pas la même d'un système à l'autre. L'autre différence concerne deux critères de jointures (voir chapitre 4.2.1.2). Il est normal que les scores de certains critères de jointures ne soient pas les mêmes d'un système à l'autre, car les règles de jointures dans la base de connaissances ont été modifiées par rapport au système PDG.

Pour ce qui est des résultats de la deuxième étape de test, que l'on retrouve dans le même tableau, on constate également qu'il n'y a aucune divergence entre les résultats des deux systèmes. En effet, les critères d'évaluation faible identifiés sont les mêmes pour les deux systèmes.

Finalement, la troisième étape de test, celle de comparaison des recommandations produites par les deux systèmes, a donné de bons résultats. En effet, les mêmes recommandations ont été produites par les deux systèmes. Il est à noter qu'il est normal de trouver qu'une seule recommandation concernant les secteurs d'évaluation « Efficacité » et « Vulnérabilité », ces derniers portant seulement sur les résultats et sur les pratiques respectivement.

### Tests avec problèmes rencontrés

Nous venons de faire la présentation d'une itération de test qui ne comportait aucun problème et aucune divergence entre les résultats du système PDG et du système PDGII. Bien entendu, il s'agit d'une itération survenue en fin de cycle d'un test. Pour en arriver à l'obtention d'une itération de test qui ne comportait aucun problème, il a fallu faire plusieurs itérations et corriger au fur et à mesure de ces itérations, le système PDGII par rapport aux problèmes rencontrés.

Nous ne détaillerons pas ces itérations avec problèmes en détail comme nous l'avons fait précédemment dans cette section avec la présentation du tableau 5.1 de l'itération de test sans problème. La raison est que les problèmes rencontrés étaient souvent très semblables. Par conséquent, nous discuterons globalement de ces problèmes rencontrés et qui ont dû être corrigés. Voici la liste de ces problèmes à chaque étape de test.

- **Problèmes de la première étape de test :** Divergence dans les scores de performance des critères d'évaluation. Il s'agit du problème qui est survenu le plus souvent et qui est le plus important. Puisque les scores de performance des critères d'évaluation sont la base de l'évaluation de la performance d'entreprise, une mauvaise évaluation du score d'un critère a des répercussions sur les

résultats des deux autres étapes de test. Par conséquent, à chaque divergence de scores de critères du système PDGII par rapport au système PDG, le problème a dû être trouvé et corrigé. Comme le calcul de ces scores se fait à l'aide de règles et de procédures contenues dans la base de connaissances, toutes les fois que cette problématique est survenue étaient causées par des erreurs d'implémentation dans la base de connaissances. Ces erreurs furent facilement corrigées grâce à la bonne structuration de cette dernière. Mises à part les erreurs d'implémentation dans la base de connaissances, quelques différences sur la précision des décimales dans les calculs provoquaient certaines divergences de scores, ce qui fut aussi rapidement corrigé.

- **Problèmes de la deuxième étape de test :** Divergence dans l'identification des critères les plus faibles. L'identification des critères les plus faibles a posé problème seulement lorsqu'il y avait des erreurs dans la première étape de test.
- **Problèmes de la troisième étape de test :** Divergence dans les recommandations produites. Des divergences de recommandations produites sont survenues seulement lorsqu'il y avait des erreurs dans la première étape de test.

## 5.2 Tests et validation des améliorations futures

Les tests qui ont été effectués et présentés dans la section précédente de ce chapitre concernaient les tests de validation pour une première version complète et fonctionnelle de la base de connaissances. Une fois cette base de connaissances bien testée et validée, elle peut être utilisée pour la production de diagnostic.

Cependant, comme le but premier du LaRePE est de développer son expertise sur le diagnostic de performance d'entreprise, il est certain qu'avec le temps, l'expertise se trouvant dans la base de connaissances sera sujette à des améliorations et mises à jour. De plus, le système PDG ne pourra plus être utilisé pour appliquer les tests, étant donné qu'il s'agira de nouveaux éléments d'expertise. Par conséquent, il faut adopter un nouveau mode de fonctionnement de test qui pourra être utilisé lors de modifications aux éléments d'expertise contenus dans la base de connaissances.

Tout d'abord, le laboratoire se doit d'utiliser un environnement de développement. Pour l'instant, une base de données de développement est utilisée afin de faire différents tests lorsque nécessaire. Les modifications et les tests sur cette base de données n'ont aucune répercussion sur la base de données officielle de production. Il en va de même pour la base de connaissances. Il faut posséder une base de connaissances de développement, copie de celle utilisée en production, consacrée aux tests des modifications qui doivent être appliquées. Par conséquent, lorsqu'une modification doit être appliquée sur un élément d'expertise de la base de connaissances, elle serait appliquée à la base de

connaissances de développement. Par la suite, les tests nécessaires s'en suivraient afin de valider les nouveaux résultats.

Une fois ces nouveaux résultats obtenus, il n'y a qu'une seule façon de les valider. L'expert humain, d'où provient l'expertise contenue dans la base de connaissances, est le mieux placé pour valider les résultats issus de la base de connaissances. Par conséquent, un des experts humains du laboratoire se devra de vérifier et valider tous les résultats issus de modifications apportées à la base de connaissances.

Finalement, lorsque ces nouveaux résultats auront été vérifiés et validés par l'expert humain, les modifications pourront alors être apportées à la base de connaissances officielle de production. Bien entendu, les mêmes tests devront être apportés afin de s'assurer que l'on obtient les mêmes résultats qu'avec l'environnement de développement de la base de connaissances.

La figure 5.2 illustre les propos précédents.

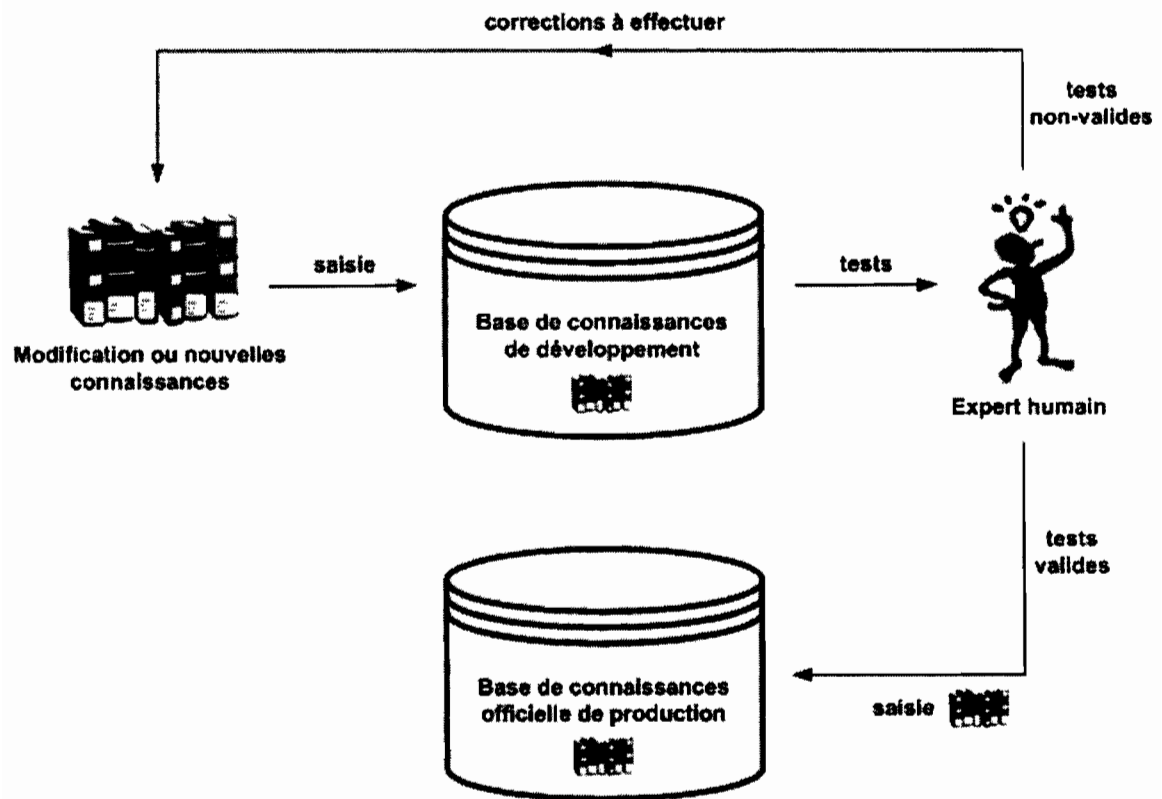


Figure 5.2 Processus de maintenance de la base de connaissances

## 5.3 Processus complet d'un diagnostic PDGII

Après toutes ces discussions sur le nouveau système PDGII, revoyons maintenant en détail le processus complet de la production d'un diagnostic, débutant avec le questionnaire que l'entreprise doit remplir jusqu'à la production du rapport final des résultats de diagnostic de la performance de l'entreprise.

Voici les différentes étapes du processus de production d'un diagnostic avec le système PDGII:

1. Questionnaire (et états financier) à remplir par l'entreprise, provenant d'un intermédiaire ou du Groupement des chefs d'entreprise.
2. Questionnaire (et états financier) rempli et envoyé à un intermédiaire ou au Groupement des chefs d'entreprise, qui le renvoi au LaRePE en censurant le nom de l'entreprise (pour des raisons de confidentialité).
3. Réception, validation et saisie du questionnaire (et des états financier) dans l'entrepôt de données du LaRePE par un membre de l'équipe du LaRePE.
4. Configuration du diagnostic à l'aide du programme utilisateur du système PDGII (voir chapitre 4.2.2).
5. Exécution du diagnostic par le système.
6. Présentation des résultats par le système avec possibilité de modification des résultats (voir chapitre 4.2.2).
7. Configuration du rapport de résultats à l'aide du programme utilisateur du système PDGII (voir chapitre 4.2.2) et ajout manuel des éléments de recommandation du troisième niveau d'expertise (voir 4.2.1.1).
8. Production du rapport PDF par le système PDGII (voir 4.2.4).

Nous ne présenterons pas en détail les écrans du programme utilisateur pour chacune des étapes de productions d'un diagnostic de performance d'entreprise puisque cela a déjà été fait dans le chapitre 4.2.2.1.

Pour ce qui est du rapport de résultats du diagnostic, il ne sera pas non plus présenté au complet. Étant donné que le système PDGII n'est pas encore en utilisation, la mise en pages du rapport du système PDGII n'est pas finale. Par contre, comme nous l'avons vu dans le chapitre 4.2.4.2, la programmation de base du nouveau rapport générique a été effectuée et nous avons été en mesure de produire une version préliminaire du rapport du système PDGII. Voici dans les figures 5.3 et 5.4, deux exemples de page du rapport de résultats de diagnostic du système PDGII.

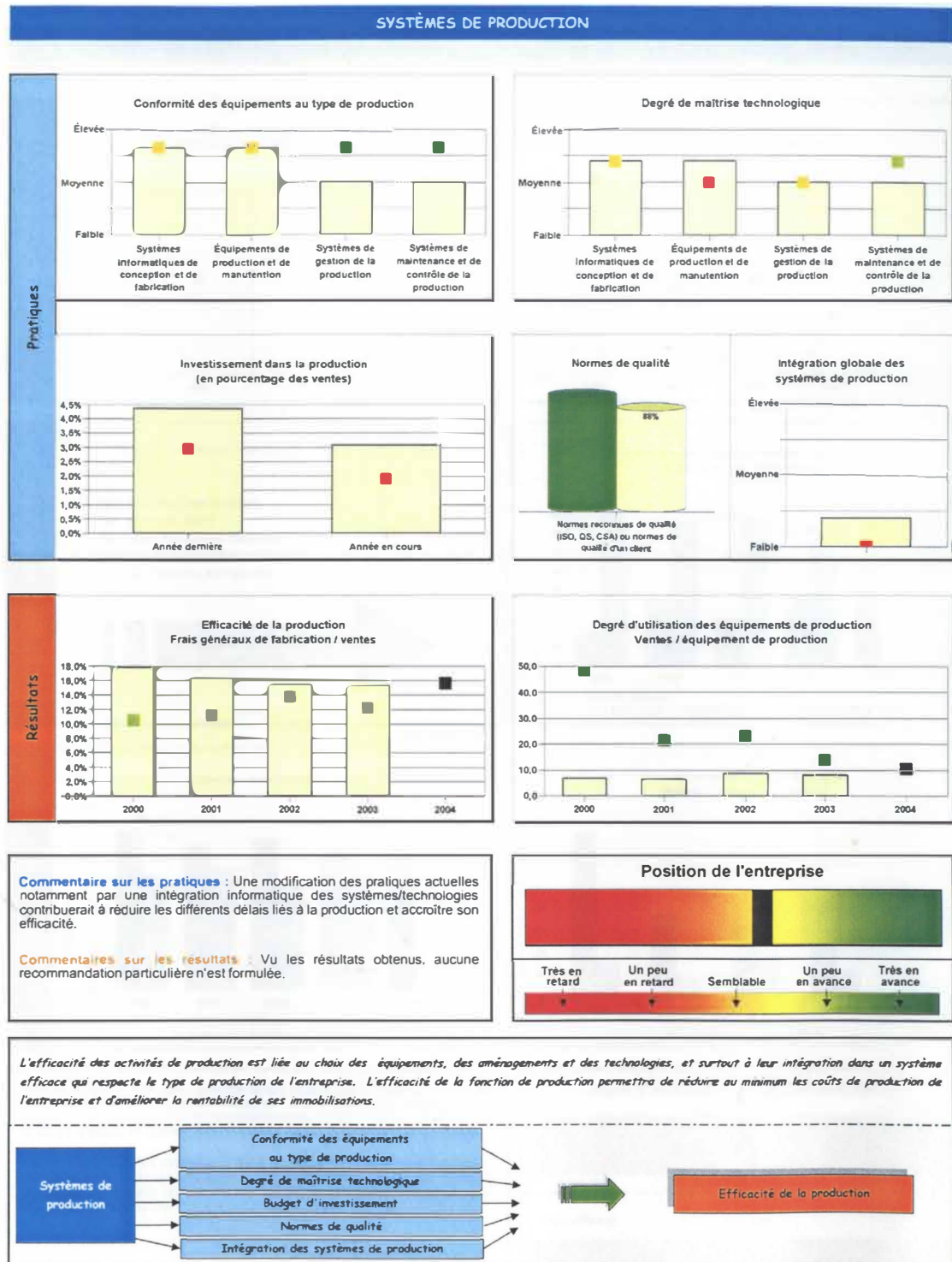


Figure 5.3 Exemple d'une page de résultats du rapport PDGII.

## EFFICACITÉ

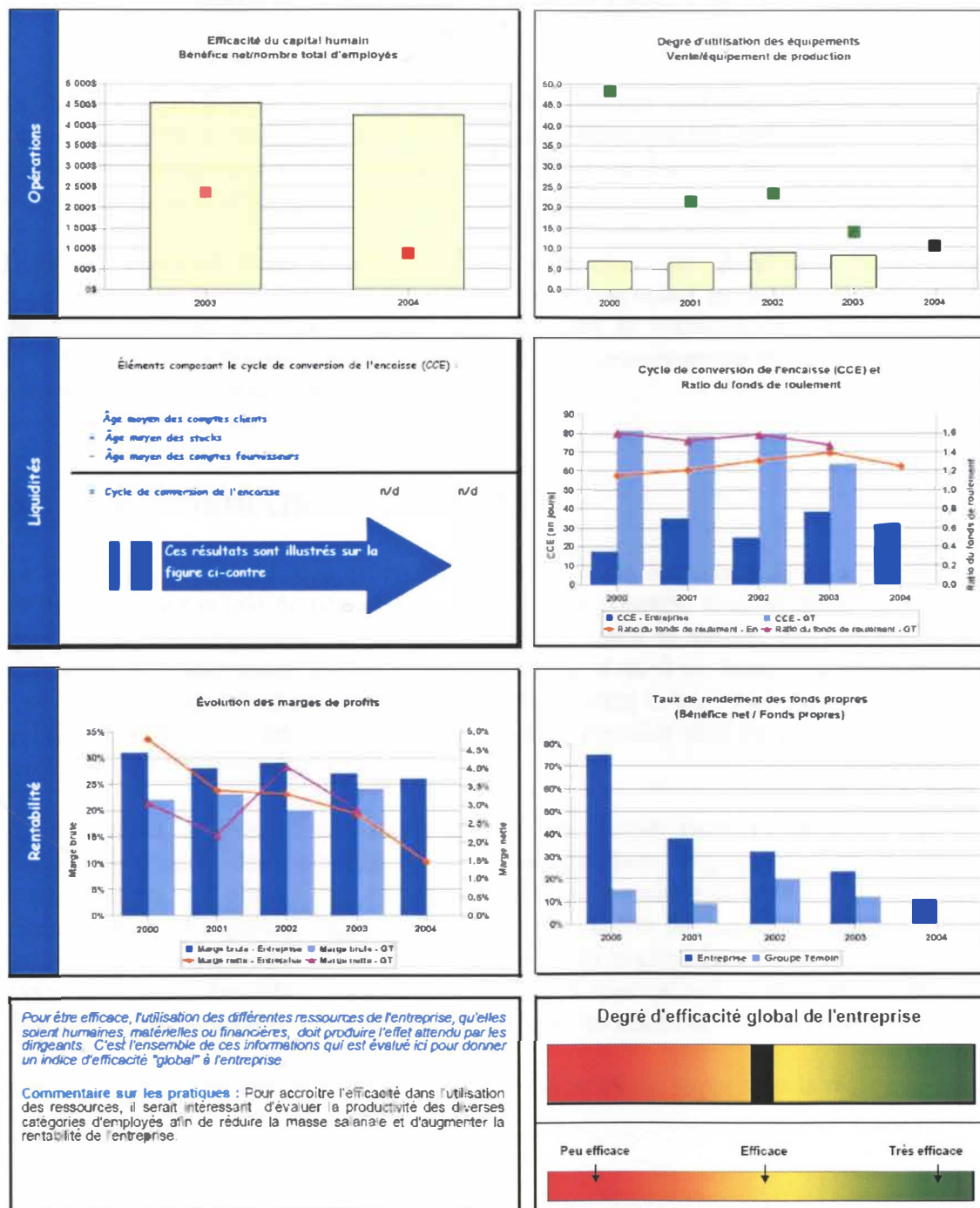


Figure 5.4 Exemple d'une page de résultats du rapport PDGII.

## CHAPITRE 6

### AMÉLIORATIONS POUR LE FUTUR

Le système PDGII et ses solutions sont venus régler une problématique qui régnait au sein du LaRePE et de son système PDG original. Bien que les améliorations apportées par le nouveau système PDGII soient notables, il existe d'autres avenues qui pourraient être explorées pour d'éventuelles améliorations.

Dans ce chapitre, nous étudierons deux approches qui pourraient faire parties d'améliorations futures. Nous étudierons ces deux approches en détail avant d'explorer de quelle manière elles pourraient se joindre dans le contexte du nouveau système PDGII. Il s'agit du **traitement de l'incertitude** des connaissances et de l'intégration des connaissances en une **ontologie** complète.

#### 6.1 Traitement de l'incertitude

Le traitement de l'incertitude est un élément qui n'a pas été touché jusqu'à présent dans ce projet. Avant même de penser à travailler sur ce concept, nous voulions être assurés que la base de connaissances soit dans un état avancé et validé. Étant donné que le traitement de l'incertitude est un travail rigoureux et qu'il ne fallait pas travailler pour rien, il était préférable d'attendre que les éléments dans la base de connaissances soient solides et qu'ils aient fait leur preuve, avant d'y travailler plus en profondeur avec du traitement de l'incertitude.

Le traitement de l'incertitude sert à prendre en compte toute l'incertitude que peut comporter chacun des éléments de connaissance (faits, règles, etc.) de la base de connaissances. La plupart des systèmes experts sont construits dans l'optique où tous leurs éléments d'expertise (faits, règles) sont vrais, aucun facteur d'incertitude n'est pris en compte. L'incertitude est un aspect trop souvent négligé lors de la conception d'un système expert. En effet, selon les domaines d'applications, l'incertitude peut être facteur assez important et peut même venir biaiser les résultats.

Pour mieux comprendre ce que l'on entend par incertitude, supposons la règle suivante :

➤ *si A et B alors C*

Il y a une notion d'incertitude nécessairement associée à une règle de ce genre. Cette incertitude peut se retrouver de 3 façons différentes :



- Il peut y avoir une incertitude quant à la validité des valeurs des données A et B.
- Il peut y avoir une incertitude quant à la validité de la règle. Est-ce que le fait d'avoir A et B implique C dans absolument tous les cas ?
- Il peut y avoir une incertitude encore plus générale sur les concepts A, B et C de la règle (domaine de la logique floue).

Les deux premiers cas d'incertitude, peuvent se régler à l'aide de techniques issues du monde des probabilités. Bien entendu, on ne peut supprimer à 100% l'incertitude que peuvent comporter les données et les règles d'un système, par contre, l'utilisation de techniques probabilistes vient grandement améliorer la validité des résultats. Le dernier cas, quant à lui, peut se régler à l'aide de la logique floue.

### L'incertitude dans le système PDGII

En étudiant bien le système PDGII, on se rend compte qu'il existe bel et bien des éléments d'incertitude et celle-ci se situe à deux niveaux.

- Au niveau des données sources des entreprises sur lesquelles le diagnostic est exécuté.
- Au niveau de l'expertise même du système.

Débutons par l'incertitude au niveau des données sources. Malgré l'amélioration du système PDGII par rapport au système PDG quant à son automatisation, le nouveau système n'est pas entièrement automatisé. En effet, le processus de production d'un diagnostic PDGII nécessite encore des interventions humaines. Les deux interventions humaines les plus importantes se situent au tout début du processus, soit lorsque l'entreprise remplit le questionnaire papier d'une vingtaine de pages et lors de la saisie des données de ce même questionnaire dans la base de données.

Le fait que l'entreprise ait à remplir un questionnaire papier apporte un degré d'incertitude au système. Comme le questionnaire est assez exhaustif et qu'il contient un grand nombre de questions, il est normal que des erreurs se glissent dans les réponses. Ces erreurs peuvent provenir autant d'une mauvaise compréhension d'une question que d'une simple erreur d'écriture dans la réponse. Ce type d'incertitude est très difficile à régler. Par contre, afin de diminuer les risques d'erreurs et ainsi réduire au maximum l'incertitude, une grille d'explication est fournie avec le questionnaire pour expliquer chaque question et aussi indiquer où se trouvent les données pertinentes.

D'un autre côté, l'étape de saisie des données du questionnaire dans la base de données par un assistant de recherche du laboratoire comporte aussi son lot d'incertitude. Cette intervention humaine peut venir glisser d'autres erreurs dans les données de l'entreprise.

En effet, avec le grand nombre de données qu'un assistant de recherche doit entrer dans l'ordinateur pour un seul questionnaire, il est plus que possible qu'il fasse des erreurs dans la saisie. Afin de diminuer les possibilités d'erreurs et ainsi réduire l'incertitude dans ce cas, une contre-vérification est faite pour chaque questionnaire saisi.

Pour le restant de ce chapitre, nous nous attarderons sur l'incertitude qui règne au niveau des éléments d'expertise contenus dans la base de connaissances. Nous verrons que cette incertitude peut être cernée à l'aide de différentes techniques probabilistes et de logique floue. Avant de voir en détail ces techniques, identifions lesquelles sont à notre disposition.

### Technologie existante

La compagnie LPA, d'où provient le logiciel Flex, possède un logiciel servant spécifiquement au traitement de l'incertitude. Il s'agit du logiciel Flint [17] et permet de faire du traitement de l'incertitude directement sur les bases de connaissances Flex.

Flint permet l'utilisation de trois techniques différentes, l'une étant une technique de logique floue et les deux autres étant des techniques probabilistes :

- La mise à jour Bayésienne (« *Bayesian updating* »)
- Les facteurs de certitude (« *Certainty factors* »)

Analysons chacune de ces 3 techniques afin de se faire une meilleure idée des possibilités offertes par Flint pour le système PDGII.

#### **6.1.1 Mise à jour Bayésienne (« *Bayesian updating* »)**

Pour prévoir les choses, on peut se baser sur les probabilités. Par exemple, on peut savoir qu'environ 45% des gens d'une petite ville sont de sexe masculin. Cependant, cette probabilité peut être appelée à changer en présence de certains facteurs importants. Ici dans notre exemple, on pourrait apprendre qu'un important tournoi de golf professionnel masculin a lieu dans la ville. Ce nouveau fait viendra certainement modifier à la hausse la probabilité de 45% d'être de sexe masculin dans cette ville pendant la durée du tournoi.

La révision de probabilités est un principe fondamental de la mise à jour Bayésienne. Cette technique de révision de probabilité ne serait pas possible sans le théorème de Bayes [17, 23]. Ce théorème provient du mathématicien Thomas Bayes et la formule qui en est à la base est la suivante :  $P(A|B) = P(B|A) * P(A) / P(B)$

Cette formule nous permet de voir l'influence qu'ont les événements entre eux, plus précisément la probabilité que l'événement A survienne sachant que l'événement B est vrai.

Reprenons notre exemple de probabilité d'être de sexe masculin dans notre ville. Identifions les probabilités suivantes :

- $P(A) = 0.45$ , la probabilité pour un individu dans la ville d'être de sexe masculin.
- $P(B) = 0.000001$ , la probabilité pour un individu d'être un joueur de golf professionnel.
- $P(B|A) = 0.000002$ , la probabilité pour un individu d'être un joueur de golf professionnel sachant qu'il est de sexe masculin.

En appliquant la formule de Bayes, on peut réviser la probabilité d'être de sexe masculin dans cette ville sachant que l'individu est un joueur professionnel de golf.

$$P(A|B) = P(B|A) * P(A) / P(B) = 0.000002 * 0.45 / 0.000001 = 0.891$$

En rajoutant une condition, celle d'être un joueur de golf professionnel, la probabilité d'être de sexe masculin est grimpée de 45% à 90%.

La mise à jour Bayésienne consiste donc à incorporer une à la suite des autres les différentes évidences de révision de probabilité à l'aide du théorème de Bayes.

Avec Flint, il est possible d'utiliser ce principe en construisant un réseau de probabilités Bayésienne. Chaque règle du système, accompagnée d'une probabilité, se retrouvera dans ce réseau de probabilités. Pour utiliser ce réseau, il suffit d'entrer les données de la situation dans le réseau. Les probabilités concernées seront combinées et une probabilité finale, que l'événement à vérifier survienne, sera calculée.

L'utilisation de cette technique pour le traitement de l'incertitude comporte plusieurs avantages :

- La technique est basée sur un théorème statistique prouvé (théorème de Bayes).
- Les évidences sont exprimées à l'aide de probabilité.
- Le poids des règles est basé sur leur probabilité d'évidence.

Par contre, cette technique possède aussi des désavantages :

- Comme il s'agit d'une technique qui révisé les probabilités de faits ou de règles lorsqu'ils sont conditionnés par d'autres faits ou règles, on se doit absolument de connaître les probabilités de départ, ce qui en pratique n'est pas toujours simple.

- Comme il peut exister des dépendances dans les probabilités, il faut être en mesure de calculer ou d'estimer ces dépendances.
- La valeur d'une probabilité ne nous informe en rien sur son niveau de justesse.
- L'ajout de nouvelles règles nécessite bien souvent la modification des probabilités de départ de d'autres règles dans le système, ce qui peut devenir un exercice lourd.

### 6.1.1.1 La mise à jour Bayésienne et le système PDGII

Voyons maintenant globalement à un niveau plus technique de quelle manière Flint peut être utilisé pour implémenter la technique de mise à jour Bayésienne, afin de faire la gestion de l'incertitude dans une base de connaissances Flex.

Voici ici l'exemple d'une règle Flex, comme nous en avons vu précédemment dans ce document, mais qui contient le mot réservé **uncertainty\_rule** pour lui permettre de faire du traitement de l'incertitude.

```
uncertainty_rule rule01
    if the temperature is high
    and the water_level is not low
    then the pressure is high .
```

Cette règle n'exécute cependant pour l'instant aucun traitement de l'incertitude.

Pour ce faire, nous pouvons entre autres attacher des poids de confiance aux règles. Ces poids (« affirms » et « denies ») reflètent la confiance que nous avons dans les affirmations (nouvelles évidences) qui se trouvent dans une règle. Par exemple, plus le poids « affirms » est élevé, plus le niveau de confiance est élevé sur cette affirmation. Une règle Bayésienne dans le langage Flex a donc l'air de ceci :

```
uncertainty_rule rule01_bayesian
    if the temperature is high ( affirms 18.00 ; denies 0.11 )
    and the water_level is not low ( affirms 1.90 ; denies 0.10 )
    then the pressure is high .
```

Par le même principe, en utilisant une valeur « affirms » et « denies » à 1, avec une probabilité P « the pressure is high » à 0.5, nous obtenons une règle équivalente à la règle sans traitement d'incertitude « rule01 » :

```
uncertainty_rule rule01_bayesian
    if the temperature is high ( affirms 1.00 ; denies 1.00 )
    and the water_level is not low ( affirms 1.00 ; denies 1.00 )
    then the pressure is high .
```

Enfin, les règles avec probabilités peuvent être appelées à l'aide de cette relation qui contient les probabilités de départ (0.03 et 0.98).

```
relation simple_boiler_probability( P )  
    if trace propagation  
    and reset all probability values  
    and the probability that the water_level is low = 0.03  
    and the probability that the temperature is high = 0.98  
    and propagate simple_boiler_control probability rules  
    and the probability that the pressure is high = P .
```

C'est ensuite à l'aide des poids de confiance « affirms » et « denies » contenus dans les règles que les probabilités de départ seront revus à la hausse ou à la baisse selon le cas.

Pour revenir au niveau du système PDGII, dans le cas particulier des PME, la règle voulant que le passé soit garant de l'avenir est peu pertinente. En effet, les PME sont bien souvent vulnérables à leur environnement externe et selon leur contexte, elles peuvent être plus ou moins fragiles. Par conséquent, l'utilisation de la technique de mise à jour Bayésienne n'est peut-être pas celle qui doit être privilégiée dans le cas du système PDGII.

### **6.1.2 Facteurs de certitude (« *Certainty factors* »)**

Les facteurs de certitude [17, 23] tentent de palier aux différentes imperfections de la technique Bayésienne. Pour ce faire, elle utilise des facteurs de certitude au lieu d'utiliser des probabilités sur les faits et règles du système traité.

Un facteur de certitude peut avoir une valeur située entre -1 et 1. Une règle ayant un facteur de certitude de 1 a une probabilité de 100% d'être vraie, tandis qu'une règle avec un facteur de certitude de -1 a une probabilité de 100% d'être fausse. Une valeur de facteur de certitude à 0 représente une certitude neutre, inconnue et donc une incertitude totale.

Cette approche a été développée pour le système expert de diagnostic médical MYCIN [12, 20, 23] (voir chapitre 3.2.4) et a été largement utilisée pour le traitement de l'incertitude des systèmes experts dans la fin des années 1970 et dans les années 1980.

#### **6.1.2.1 Les facteurs de certitude et le système PDGII**

Voyons maintenant globalement de quelle manière il est possible d'utiliser les facteurs de certitude avec Flint.

Prenons par exemple la règle suivante :

```
uncertainty_rule rule02
    if the release_valve is stuck
    then the release_valve is need_cleaning .
```

Cette règle ne fait aucun traitement de l'incertitude. Voici maintenant de quelle manière avec Flint on peut réécrire cette règle pour y ajouter un aspect pour le traitement de l'incertitude à l'aide des facteurs de certitude :

```
uncertainty_rule rule02_cf
    if the release_valve is stuck
    then the release_valve is need_cleaning
    with certainty factor 0.76 .
```

On voit qu'il n'y a pas beaucoup de différences au niveau de la syntaxe entre la première règle (pas de traitement de l'incertitude) et la deuxième règle (traitement de l'incertitude). Il est donc relativement facile de créer un système sans traitement de l'incertitude pour ensuite lui ajouter les éléments reliés au traitement de l'incertitude.

Enfin, cette relation servira à appeler les règles avec facteurs de certitude :

```
relation boiler_cf( CF1, CF2, CF3 )
    if trace propagation
    and reset all certainty_factor values
    and the certainty_factor that the water_level is low = -1
    and the certainty_factor that the warning_light is on = 1
    and propagate boiler_control certainty_factor rules
    and the certainty_factor that the release_valve is stuck = CF2
    and the certainty_factor that the release_valve is need_cleaning =
                                                CF3
```

Cette technique qui semble intéressante à prime abord, comporte cependant certaines limites qui sont même reconnues par son inventeur [23].

### 6.1.3 Logique floue (« *Fuzzy logic* »)

La logique floue [17, 23] sert à raisonner avec les descriptions vagues qui font souvent partie d'éléments de connaissance. Par exemple, prenons les faits suivants :

- Jacques est grand.
- Un médecin dit que son patient souffre d'une très forte fièvre.

Est-ce vrai de dire que Jacques est grand, sachant qu'il mesure 5 pieds 11 pouces ? Si l'on pose la question à plusieurs personnes, il se peut que les réponses soient différentes puisque les critères de grandeur ne sont pas les mêmes pour tout le monde. Une de ces

personnes pourrait par exemple répondre que oui, qu'il s'agit bien d'une grande personne. Cela voudrait-il dire qu'une personne mesurant 5 pieds 10<sup>3</sup>/<sub>4</sub> pouces est petite? De plus, qu'est-ce qu'une très forte fièvre, 39°C, 41°C ? Ces faits contiennent des descriptions vagues qui demandent plus de précision, c'est là qu'intervient la logique floue.

En fait, on peut se rendre compte qu'il ne s'agit pas d'incertitude au niveau de la vérité du fait. La grandeur ou la température d'une personne sont des faits véritables. La notion d'incertitude sur ce genre de fait concerne plus le caractère vague de sa description, au niveau linguistique. Par conséquent, plusieurs auteurs du domaine s'entendent pour dire que la logique floue n'est pas une technique pour faire le traitement de l'incertitude présente dans le raisonnement sur une base de connaissances. Mais tous s'entendent pour dire que la logique floue est une technique permettant de rendre moins floues et plus précises les différentes connaissances d'une base de connaissances.

À part les descriptions vagues dont peuvent faire l'objet divers éléments de connaissance, la logique floue peut venir aider les règles de systèmes experts traditionnels fonctionnant bien souvent à l'aide de seuils préétablis et exacts. Par exemple, prenons les antécédents des règles suivantes :

- If the person's temperature  $\leq 37^{\circ}\text{C}$  then no fever
- If the person's temperature  $> 37^{\circ}\text{C}$  and  $< 38^{\circ}\text{C}$  then the fever is medium.
- If the person's temperature  $\geq 38^{\circ}\text{C}$  and  $< 40^{\circ}\text{C}$  then the fever is high.
- If the person's temperature  $\geq 40^{\circ}\text{C}$  and  $< 42^{\circ}\text{C}$  then the fever is very high.

Ce genre d'antécédent amène bien souvent à de l'imprécision dans le conséquent de la règle. Ces antécédents sont fixés sur des valeurs de température trop fixes, trop précises. En effet, quelle différence y a-t-il entre une valeur de 39.9°C et 40°C ? Selon les antécédents précédents, il y aura une différence dans le conséquent pour une différence de 0.1°C de température, ce qui n'est pas appréciable.

Dans ce genre de situation, la logique floue tenterait de remplacer les valeurs numériques fixes contenues dans les règles par des valeurs quantitatives non-fixes par l'utilisation de seuils ( $\leq 37^{\circ}\text{C} \pm 0.2^{\circ}\text{C}$ ). Par la suite, on pourrait associer ces valeurs quantitatives non-fixes avec seuils par des valeurs qualitatives pour simplifier la syntaxe des antécédents :

- If the person's temperature is correct
- If the person's temperature is medium
- If the person's temperature is high
- If the person's temperature is very high

### 6.1.3.1 La logique floue et le système PDGII

Voyons maintenant les éléments du système PDGII qui seraient susceptibles d'avoir besoin de la logique floue et de quelles manières Flint applique sa logique floue.

On retrouve en effet des éléments de connaissance du système PDGII sur lesquels on pourrait faire l'utilisation de la logique floue. Ces éléments se situent surtout au niveau des règles d'attribution des scores de performance des critères d'évaluation. Comme nous l'avons vu dans le chapitre 4.2.1.2, chaque critère d'évaluation se voit attribuer un score de performance allant de 0 à 4 (après une moyenne de plusieurs sous-scores de 0 à 4 pour chacun des critères). Un peu comme dans l'exemple de fièvre de la section précédente, les règles d'attribution de ces scores de 0 à 4 font l'utilisation d'intervalles fixes. Par exemple, voici 5 règles permettant d'attribuer des scores de 0 à 4, où GT représente la valeur du groupe témoin, et E représente la valeur de l'entreprise (l'ordre des règles est important).

- if GT > 0 and E > 0 and E/GT ≤ (0.5) then Score = 0
- if GT > 0 and E > 0 and E/GT ≤ (0.9) then Score = 1
- if GT > 0 and E > 0 and E/GT < (1.1) then Score = 2
- if GT > 0 and E > 0 and E/GT < (1.5) then Score = 3
- if GT > 0 and E > 0 and E/GT ≥ (1.5) then Score = 4

Comme on peut le constater, la graduation des scores est de 1, et par conséquent aussi petite que peut être la différence entre deux valeurs, les scores peuvent différer de 1, ce qui peut avoir parfois des conséquences sur les résultats.

Par exemple, si nous avons des valeurs de GT et de E plus grandes que 0 et un ratio E/GT de 0.9, alors nous obtiendrons un score de 1. Par contre, une différence dans le ratio de 0.01 plus bas viendra changer le score à 0. Cette situation peut se produire assez souvent, vu le nombre important de critères d'évaluation ainsi que le nombre important de ces règles dans le diagnostic de performance d'entreprise. Avec l'expérience, on se rend compte que cette situation n'a pas trop de conséquences graves sur l'évaluation d'une entreprise, par contre, par souci de précision, il serait peut-être utile de revoir l'implémentation de ces règles afin d'utiliser Flint et sa logique floue.

Voyons maintenant rapidement l'outil Flint et sa technique de logique floue.

Essentiellement, il y a deux composantes principales dans l'utilisation de la logique floue avec Flint :

- *Les variables floues.* Une variable floue est représentée par un intervalle de valeurs numériques. Cet intervalle numérique est alors divisé en sous-intervalle et chacun de ces intervalles est associé à un qualificateur. Ce qualificateur est à son tour représenté par un terme qualificatif linguistique et par une fonction



d'appartenance qui calcule le degré d'appartenance du qualificateur au contexte de départ.

- *Les règles floues.* Une règle floue réfère à un ou plusieurs qualificateurs d'une variable floue dans l'antécédent, et réfère à un seul qualificateur d'une variable floue dans son conséquent. La règle calcule le degré d'appartenance de chacun des qualificateurs de variable présent dans son antécédent, et calcule ensuite le nouveau degré d'appartenance du qualificateur unique de variable présent dans son conséquent.

Le processus de la logique floue avec Flint s'exécute en 3 étapes.

- « *Fuzzification* ». Dans cette étape, les valeurs de départ en entrée seront associées à des variables floues et le degré d'appartenance de chacun des qualificateurs de ces variables floues sera calculé, degré d'appartenance par rapport aux données en entrée bien sûr.
- *Propagation.* Les règles en rapport avec les variables floues assignées lors de la première étape seront appliquées. C'est ainsi que les qualificateurs dans le conséquent de ces règles se verront attribuer un nouveau degré d'appartenance.
- « *De-fuzzification* ». Les valeurs de degré d'appartenance des qualificateurs en cause seront reconverties en valeurs de sorties.

Voici dans la figure 6.1, un exemple des 3 étapes du processus de logique floue. Les informations en entrée sur une turbine sont les suivantes : la température est à 200° et la pression est à 15Kpa.

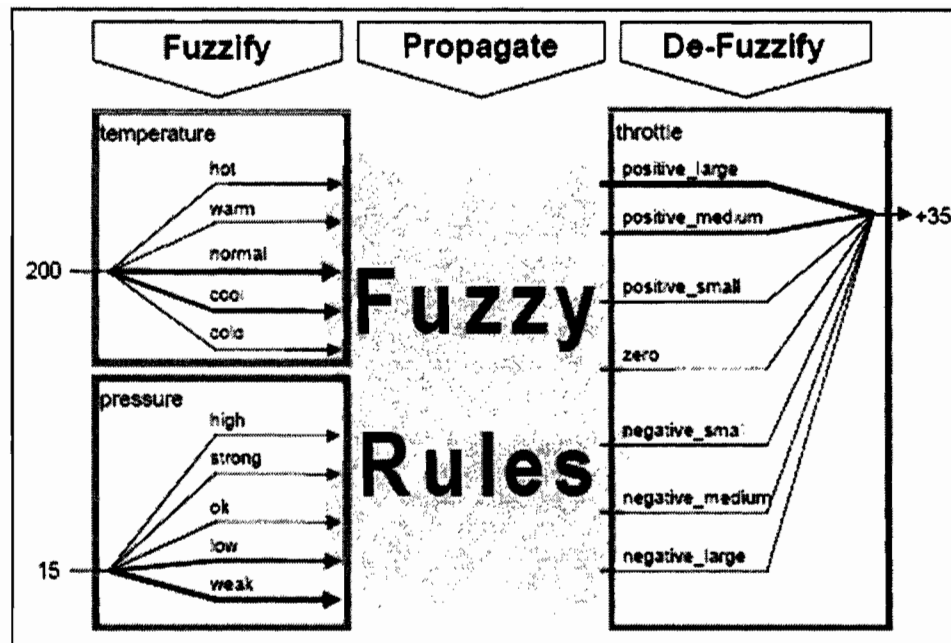


Figure 6.1 Exemple des 3 étapes de logique floue avec Flint. [17]

Dans la première étape, le qualificateur de la variable floue « temperature » ayant le degré d'appartenance le plus élevé est « normal » suivi de « cool ». D'un autre côté, le qualificateur de la variable floue « pressure » ayant le degré d'appartenance le plus élevé est « weak » suivi de « low ».

En deuxième étape, il y a propagation des règles et à la sortie de cette étape, un degré d'appartenance pour tous les qualificateurs de la variable « throttles » a été calculé et nous constatons que c'est le qualificateur « positive\_large » qui possède le degré d'appartenance le plus élevé pour la variable floue « throttles » suivi du qualificateur « positive\_medium ».

En troisième et dernière étape, il y a utilisation de tous les degrés d'appartenance des qualificateurs de la variable floue « throttles » pour le calcul d'une valeur de sortie. Cette valeur est à +35. Cette valeur aura des répercussions selon le système de turbine.

Nous n'irons pas plus en détail dans l'explication de l'utilisation de Flint et de sa logique floue car il s'agit d'une technique assez complexe. Il s'agit cependant d'une technique très intéressante pour le futur du système PDGII. Pour plus de détail sur le traitement de l'incertitude avec Flint, voir [17].

## 6.2 Ontologie

Le terme « ontologie » existe depuis environ 15 ans dans le domaine de l'intelligence artificielle. Plusieurs définitions d'une ontologie circulent maintenant dans la littérature [1, 3, 10], mais voici une des plus reconnues en informatique :

« Une ontologie est une spécification explicite d'une conceptualisation ».[10]

Une ontologie est donc la représentation et la formalisation des connaissances sur un domaine plus ou moins spécifique. Elle est le produit entre autres du travail de modélisation et de représentation fait lors du processus d'ingénierie des connaissances. Bien que le travail d'ingénierie des connaissances, tel que nous l'avons vu au chapitre 3, soit un bon départ de conception d'une ontologie, un travail supplémentaire est nécessaire pour l'obtention d'une ontologie complète.

Une ontologie est composée de la spécialisation des concepts du domaine de la base de connaissances, des différentes règles utilisant ces différents concepts, et des liens qui existent entre ces éléments (voir figure 6.2).

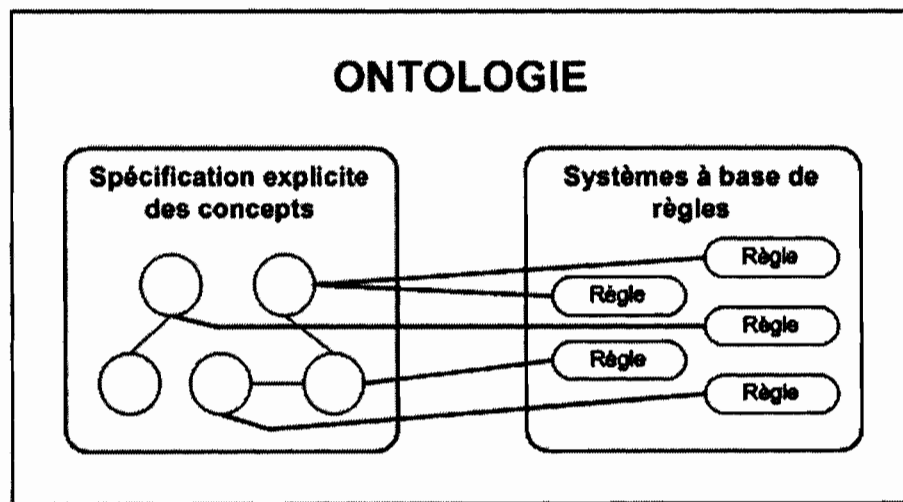


Figure 6.2 Composition d'une ontologie complète

Comme nous l'avons vu dans ce document, le système PDGII possède une base de connaissances bien structurée et organisée. Bien qu'il ne s'agisse pas encore d'une ontologie complète, cette bonne structuration et organisation de la base de connaissances constitue une base importante pour l'élaboration d'une ontologie complète et spécialisée sur le domaine de la performance d'entreprise.

En effet, la structuration de la base de connaissances en deux parties distinctes (voir chapitre 4), la partie d'expertise générale et la partie d'expertise spécifique, est un élément important à considérer pour l'élaboration d'une ontologie complète. L'ontologie qui est proposée dans cette section concerne donc autant la partie générique que la partie spécifique de la base de connaissances.

D'une part, il s'agirait d'étendre et de spécialiser la partie générale à un niveau plus élevé. Pour l'instant, cette partie contient tous les critères d'évaluation existant et leur expertise concernant le diagnostic de performance d'entreprise. Ces critères d'évaluation, comme nous l'avons vu, sont généraux au domaine de diagnostic de performance d'entreprise. C'est entre autres sur ce point qu'il faudrait travailler pour développer, de manière plus exhaustive, notre ontologie sur le domaine de la performance d'entreprise.

D'autre part, il s'agirait de spécialiser la partie spécifique de la base de connaissances. Tous les concepts des trois niveaux d'expertise du système PDGII devront être détaillés davantage pour obtenir une ontologie spécifique sur le diagnostic PDGII.

Voyons maintenant en détail de quelles manières l'ontologie (générique et spécifique) de la base de connaissances pourrait être améliorées.

### Dictionnaire de variables

Plusieurs éléments pourraient servir à venir renforcer cette ontologie. Un de ces éléments est le dictionnaire de variables développé au laboratoire. Ce dictionnaire est accessible via l'intranet du laboratoire à l'aide d'un navigateur Web et sert à la recherche rapide de variables parmi une liste de plusieurs centaines de variables développées au fil des années. Ces variables sont utilisées par les chercheurs dans leurs recherches et étant donné leur grand nombre, elles devenaient difficiles d'utilisation, d'où leur centralisation dans un dictionnaire.

Via l'intranet, il est possible de faire une recherche rapide de variables de différentes manières.

- Recherche par mots-clés.
- Recherche par nom de variables.
- Recherche par questionnaire d'utilisation (version, page, question). Les variables proviennent bien souvent de questionnaires.
- Recherche par projet d'utilisation.
- Recherche par détection de variables. Il s'agit de copier-coller un texte contenant des variables potentiels. Une recherche de variables dans ce texte sera effectuée.

Voici un exemple (figure 6.3) dans lequel nous avons fait une recherche par mots-clés avec le mot « ressources humaines ».

Figure 6.3 Dictionnaire de variables : recherche de variables

Une fois la recherche effectuée, une liste de variables associée à cette recherche est affichée plus bas dans la page. Voici une partie des résultats de la recherche effectuée avec le mot-clé « ressources humaines » (figure 6.4).

Panier	Variable	Description / Interprétation
<input type="checkbox"/>	ADRES	Sources d'information consultées pour les ressources humaines et les relations de travail: conseil d'administration ou comité consultatif
<input type="checkbox"/>	ASRES	Sources d'information consultées pour les ressources humaines et les relations de travail: associations sectorielles ou gens d'affaires
<input type="checkbox"/>	ATTESUPE	Niveau de diffusion d'informations opérationnelles: les attentes des supérieurs
<input type="checkbox"/>	BARES	Sources d'information consultées pour les ressources humaines et les relations de travail: banquiers
<input type="checkbox"/>	BUDFO	Budget alloué à la formation pour l'année en cours
<input type="checkbox"/>	BURACPRO	Présence d'un programme d'accès à la propriété pour les employés de bureau
<input type="checkbox"/>	BURDETAC	Présence d'une description de tâches pour les employés de bureau
<input type="checkbox"/>	BUREVPEP	Présence d'une politique d'évaluation du rendement pour les employés de bureau
<input type="checkbox"/>	BURPAPRO	Présence d'un programme de participation aux profits pour les employés de bureau
<input type="checkbox"/>	BURPREVI	Évolution prévue du nombre d'employés de bureau au cours de la prochaine année

Figure 6.4 Dictionnaire de variables : résultat d'une recherche de variables

Il est possible de sélectionner les variables issues de la recherche afin d'en construire un panier de variables. Ce panier, qui est une sauvegarde des variables intéressant le chercheur, servira ultimement à la production d'un « dataset » (feuille de données) à

l'aide d'un l'outil qui a aussi été implémenté au LaRePE : le « Dataset Maker ». Cet outil permet de produire des feuilles de données sur des entreprises avec les variables sélectionnées. Par contre, les variables elles-mêmes et leurs informations étant ce qui nous intéresse ici, nous n'explorerons pas davantage le « Dataset Maker ». Pour plus de détail voir [7].

Ce qui nous intéresse sont les informations sur le domaine de la performance d'entreprise contenues dans les différentes variables. Tout dépendant du type de variables, différentes informations sont disponibles. Voici des écrans (figures 6.5, 6.6, 6.7) que l'on obtient en cliquant sur une variable en particulier dans la liste de variables résultant de notre recherche.

INFORMATIONS POUR LES STATISTIQUES	
Code : ATTESUPE	Nature : Général
Description : Niveau de diffusion d'informations opérationnelles: les attentes des supérieurs	Format de la variable : DIFFUCLA
Label statistique : DIFFUSION ATTENTES DES SUPÉRIEURS	
Mots clés : communication, diffusion de l'information opérationnelle, ressources humaines	
Dépendances (variables calculées qui ont besoin de ATTESUPE directement ou indirectement) : <u>DIINOPER</u> , <u>SUMDIFFU</u> , <u>PRDIFFU</u>	

Code	Description
1	PDG / CA
2	DIRECTEUR RESPO.
3	CONTREMAÎTRE
4	EMPLOYÉ
-90	VIDE

Figure 6.5 Informations pour les statistiques sur une variable

INFORMATIONS SUPPLÉMENTAIRES (VARIABLE DU QUESTIONNAIRE)	
Nom table BD: MAN_NIV_DIFF_INF	Format dans BD: NUMBER(3)
Est-ce une variable monétaire? : Non	La variable peut-elle suivre? : Non applicable

Liste des questionnaires où se retrouve la variable:

Questionnaire	Version	Section	Page	Question
Base	4.1	Ressources humaines	7	7
Mise à jour	1.1	Ressources humaines	4	6
Base	5	Ressources humaines	7	7

Figure 6.6 Informations supplémentaires sur une variable de questionnaire



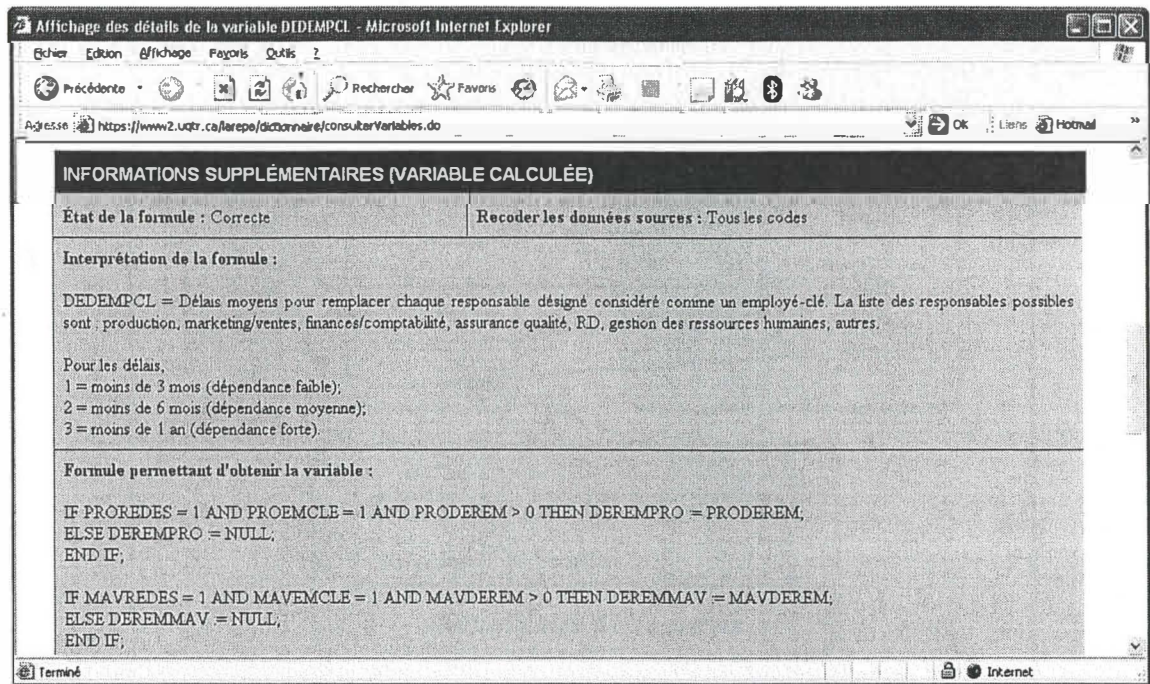


Figure 6.7 Informations supplémentaires sur une variable calculée

Les variables contiennent une quantité importante d'information pour la recherche sur la performance des entreprises. C'est d'ailleurs sur ces variables que sont basés les critères d'évaluation qui se retrouvent dans la partie générique de la base de connaissances et qui sont utilisés par le système PDGII pour le diagnostic de la performance d'entreprise. Il serait par conséquent intéressant et profitable de lier les informations de ce dictionnaire de variables aux différents concepts (critères) que l'on retrouve dans la base de connaissances, en particulier dans la partie générique de cette dernière, afin de renforcer et d'élargir l'ontologie sur le domaine (voir figure 6.8).

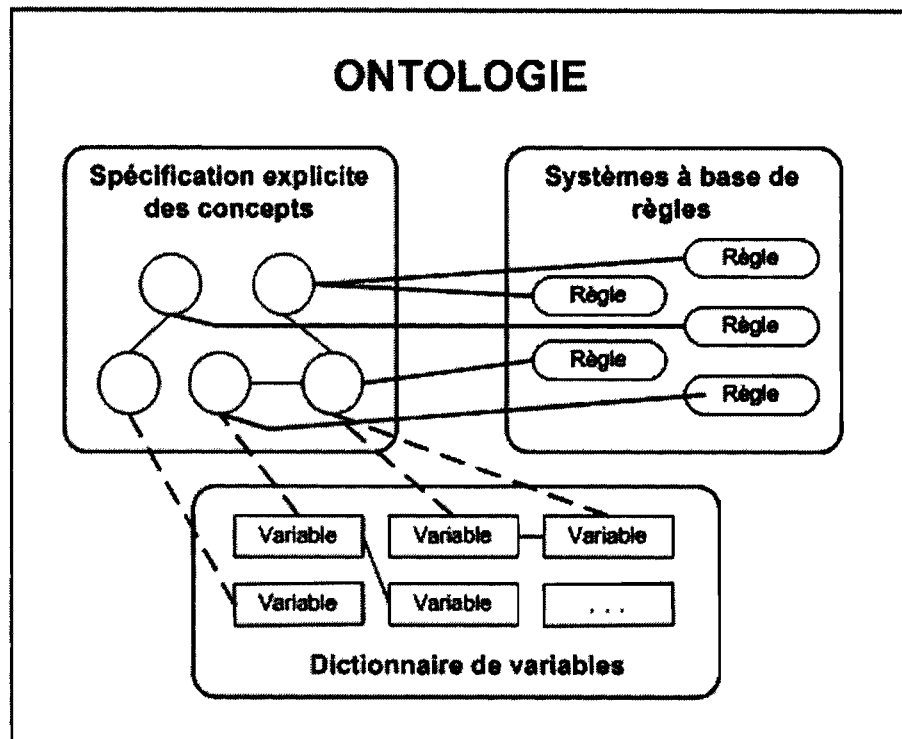


Figure 6.8 Composition de l'ontologie avec le dictionnaire de variables

Voici maintenant en détail dans le tableau 6.1, les différentes informations, issues des différents types de variables du dictionnaire, qui pourraient être reliées à la base de connaissances.

Information	Description	Exemple								
Code	Code d'identification de la variable.	ACCIIMOB								
Description	Description de la variable.	Importance accordée à l'objectif d'amélioration continue pour la diminution des accidents de travail.								
Nature	Nature de la variable.	Financier								
Format	Format de représentation des données de la variable.	Format ELEVECLA <table><thead><tr><th>Code</th><th>Description</th></tr></thead><tbody><tr><td>1</td><td>TRÈS FAIBLE</td></tr><tr><td>2</td><td>FAIBLE</td></tr><tr><td>3</td><td>MOYEN</td></tr></tbody></table>	Code	Description	1	TRÈS FAIBLE	2	FAIBLE	3	MOYEN
Code	Description									
1	TRÈS FAIBLE									
2	FAIBLE									
3	MOYEN									



		4 ÉLEVÉ 5 TRÈS ÉLEVÉ -88 PAS LA CHANCE DE RÉPONDRE -90 VIDE -99 NE S'APPLIQUE PAS
Mots clés	Liste de mots-clés représentant la variable.	amélioration continue, qualité, sécurité
Dépendance	Liste de variables qui ont besoin de la variable en question directement ou indirectement dans leur calcul.	CASTACTO, VENTACTO, STOCTO, FDRACTIF, IMMOACTI
Table BD	Table de la base de données dans laquelle on retrouve les données des entreprises sur cette variable.	NOM_TABLE
Format BD	Format de la variable dans la base de données.	NUMBER(3)
Formule	Formule permettant d'obtenir la variable.	AMOCVAEQ := AMOCU / SUMS(VAEQP,AMOCU);
Interprétation de la formule	Interprétation de la formule permettant d'obtenir la variable.	AMOCVAEQ = Amortissement cumulé / (Valeur nette des équipements de production + Amortissement cumulé)
Dépendances dans la formule	Autres variables nécessaires pour le calcul de la formule.	AMOCU, VAEQP
Condition d'exécution de la formule	Condition qui doit être respectée afin que la formule soit valide et puisse être exécutée.	IF AMOCU IS NOT NULL AND SUMS(VAEQP,AMOCU) != 0 AND SUMS(VAEQP,AMOCU) IS NOT NULL THEN CALCULER := TRUE; END IF;

**Tableau 6.1 Détails des informations sur les variables du dictionnaire de variables.**

### Les 3 niveaux d'expertise

La formalisation de l'expertise en 3 niveaux qui a été faite dans la base de connaissances du système PDGII (voir chapitre 4.2.1.1) est un élément très intéressant pour une

ontologie plus complète. En effet, ces 3 niveaux de formalisation de l'expertise constituent une base intéressante pour le développement d'une partie plus spécifique au diagnostic du système PDGII. Terminer le troisième niveau ainsi que développer davantage la formalisation de ces 3 niveaux est fortement à conseiller.

#### Résumé : une ontologie générale du domaine

L'idée de séparation de la base de connaissances en deux parties distinctes est un élément très important pour le développement de la future ontologie.

En effet, d'une part la partie générique de la connaissance aidera à la constitution d'une partie plus générique de l'ontologie sur le domaine de la performance d'entreprise. Dans cette partie, on pourrait retrouver tous les éléments et concepts que l'on retrouve actuellement dans la partie générique de la base de connaissances, soit tous les concepts reliés aux différents critères d'évaluation qu'on y retrouve. De plus, comme nous l'avons vu précédemment, le dictionnaire de variables constitue une excellente source de concepts importants à intégrer dans cette partie générique de la future ontologie complète sur le domaine.

D'autre part, la partie spécifique de la base de connaissances, qui correspond à l'expertise spécifique même du système PDGII, constitue une autre source importante de concepts importants à intégrer dans une partie spécifique de la future ontologie. Dans cette partie, on pourrait retrouver les concepts relatifs à la formalisation en 3 niveaux du diagnostic du système PDGII, niveaux préalablement développés encore plus en détail comme nous l'avons précédemment conseillé dans ce chapitre.

De manière plus globale, les deux apports d'une ontologie se situent au niveau de la représentation des connaissances et au niveau du raisonnement sur ces dernières. En effet, une ontologie permet de faire la représentation simple des concepts, souvent sous la forme d'un arbre de concepts, ainsi que la définition explicite des concepts. De plus, il est possible d'exécuter du raisonnement sur une ontologie. Cela pourrait apporter un autre aspect au niveau de « l'intelligence » implicite dans la représentation des connaissances. Bref, le développement d'une ontologie complète sur le domaine de la performance d'entreprise serait d'une très grande utilité pour le LaRePE.

## **CHAPITRE 7**

### **CONCLUSION**

Le projet du système PDGII, un système expert de diagnostic de la performance d'entreprise dont faisait l'objet ce mémoire de maîtrise, est le fruit de deux années de travail au sein du Laboratoire de recherche sur la performance des entreprises (LaRePE).

La base essentielle de ce projet était le travail de réingénierie des connaissances du système précédent, le système PDG. Les nombreux éléments d'expertise que contenait le système PDG ne faisaient pas l'objet d'une structure efficace. Les éléments d'expertise étaient plutôt dispersés un peu partout dans l'implémentation du système. Cette limite nuisait grandement à son bon fonctionnement, sa compréhension et surtout à son évolution. Le système PDG possédait d'autres faiblesses plus secondaires, mais le problème central restait au niveau des éléments d'expertise.

La solution a consisté en l'élaboration d'un nouveau système PDGII à la base duquel un important travail de réingénierie des connaissances a été accompli. Un aspect important de cette réingénierie fut le travail de représentation et de formalisation des connaissances acquises du premier système. Cette réingénierie a permis la formalisation des éléments de connaissance en 3 niveaux d'expertise concernant le diagnostic de la performance d'entreprise, et en l'élaboration d'une banque générique de critères d'évaluation de la performance d'entreprise. Le résultat de ce travail de réingénierie des connaissances fut la conception d'une base de connaissances sur le domaine de la performance d'entreprise dans laquelle on retrouve deux parties. D'une part, on retrouve la partie générique contenant les connaissances plus génériques au domaine comme la banque de critères d'évaluation. D'autre part, on retrouve la partie spécifique contenant les connaissances plus spécifiques pour des systèmes spécifiques, comme le système PDGII et ces 3 niveaux d'expertise pour le diagnostic de la performance d'entreprise.

Bien que la réingénierie des connaissances et la conception de la base de connaissances soient les éléments centraux du projet, d'autres accomplissements ont été réalisés afin de faire du système PDGII un système convivial, offrant de nombreuses possibilités, contrairement au premier système PDG. Voici les réalisations majeures pour le système PDGII :

- Réingénierie complète des connaissances du système PDG menant à la conception d'une base de connaissances possédant une bonne représentation et formalisation, donnant entre autres, un caractère générique à cette dernière.

- Conception d'un nouveau programme utilisateur offrant des possibilités de configuration de paramètres et d'expertise pour le diagnostic via le Web. Il est entre autres possible de faire la configuration de divers profils de diagnostic.
- Génération d'un rapport détaillé en couleur des résultats de diagnostic. Les divers éléments de ce rapport (graphiques, recommandations) sont maintenant configurables via le programme utilisateur Web. La mise en page de ce rapport est aussi maintenant facilement modifiable.
- Utilisation d'un entrepôt de données pour le stockage des données et paramètres d'expertise initiaux et la sauvegarde des résultats détaillés des diagnostics.

Finalement, ce nouveau système PDGII et sa base de connaissances sont venus régler les difficultés de fonctionnement, la mauvaise compréhension des connaissances, et la difficulté d'évolution du système PDG. La conception du système PDGII, faite selon les règles de l'art de l'intelligence artificielle, rend son évolution très facile. De plus elle permettra l'ajout de futurs éléments d'intelligence artificielle comme le traitement de l'incertitude et le développement d'une ontologie détaillée.

## RÉFÉRENCES

1. Bateman, J. *Ontology portal*. 2005 [cited; Available from: <http://www.fb10.uni-bremen.de/anglistik/langpro/webpace/jb/info-pages/ontology/ontology-root.htm>].
2. Beaudoin, J.-F., S. Delisle, M. Dugré, J. St-Pierre. *Reengineering the Knowledge Component of a Data Warehouse-Based Expert Diagnosis System*. Lecture Notes in Computer Science. 2005. pp.910-919.
3. Clark, P. *KBS/Ontology Projects Worldwide*. 2005 [cited; Available from: <http://www.cs.utexas.edu/users/mfkb/related.html>].
4. Delisle, S., M. Dugré, et J. St-Pierre. *Multidimensional SME Performance Evaluation: Upgrading to Data Warehousing & Data Mining Techniques*. In *International Conference on Information et Knowledge Engineering*. 2004. Las Vegas, Nevada: CSREA Press.
5. Delisle, S. et J. St-Pierre. *Expertise in a Hybrid Diagnostic-Recommendation System for SMEs: A Successful Real-Life Application*. 3029 ed. Lecture Notes In Computer Science. 2004. pp.807-816.
6. Delisle, S., J. St-Pierre, et T. Copeck. *A Hybrid Diagnostic-Advisory System for Small and Medium-Sized Enterprises: A Successful AI Application*. à paraître, 2005.
7. Dugré, M., *Conception et réalisation d'un entrepôt de données : Intégration à un système existant et étape nécessaire vers le forage de données*, Mémoire de maîtrise, Département de mathématiques et d'informatique. 2004, UQTR.
8. Dugré, M. et S. Delisle, *Le système PDG : Évaluation de l'actuel et Éléments de conception du PDG II*. Travail de recherche, Laboratoire de Recherche sur la performance des entreprises. Institut de Recherche sur les PME. 2003, UQTR.
9. Groupement des chefs d'entreprise, <http://www.groupement.qc.ca/>.
10. Gruber, T., *A Translation Approach to Portable Ontology Specification*. Knowledge Acquisition, 1993. 5: 199-220.
11. Hewett, M., *Algernon in Java*. 2002, Hewett Research, <http://algernon-j.sourceforge.net/doc/algernon-protege.html>.
12. Holsapple et Whinston, *Decision Support Systems: A Knowledge-Based Approach*. 2001: Thomson Learning.

13. Institut de recherche sur les PME, <http://www.uqtr.ca/inrpme/>.
14. Laboratoire de recherche sur la performance des entreprises, <http://www.uqtr.ca/~larepel/Larepe/>.
15. Logic Programming Associates Ltd, <http://www.lpa.co.uk/>.
16. LPA, *Flex Expert System Toolkit: Flex Reference*. 1996, London: Logic Programming Associates Ltd.
17. LPA, *Flint Reference*. 2004, London: Logic Programming Associates Ltd.
18. LPA, *Intelligence Server*. 2004, London: Logic Programming Associates Ltd.
19. LPA, *ProData Interface*. 2004, London: Logic Programming Associates Ltd.
20. Marakas, G.M. *Decision Support Systems In the 21st Century*. 2nd ed. 2003: Prentice Hall.
21. PDFLIB, <http://www.pdflib.com/>.
22. Power, D.J. *DSS Resources*. <http://dssresources.com/glossary/dssglossary1999.html>. 2005.
23. Russel, S. et P. Norvig. *Artificial Intelligence A Modern Approach*. 1995: Prentice Hall.
24. St-Pierre, J. et S. Delisle, *An Expert Diagnosis System for the Benchmarking of SMEs' Performance*. *Benchmarking—An International Journal*, to appear. 11(5-6).
25. St-Pierre, J., L. Raymond, et E. Andriambeloson. *Performance Effects of the Adoption of Benchmarking and Best Practices in Manufacturing SMEs*. In *Conference on Small Business and Enterprise Development*. 2002. The University of Nottingham (UK).
26. Stanford University, *The Protégé Ontology Editor and Knowledge Acquisition System*, <http://protege.stanford.edu/>.
27. Stefik, M., *Introduction to Knowledge Systems*. 1995: Morgan Kaufmann Publishers, Inc.
28. Stottler Henke Associates Staff, *PADAL helps US Navy aircraft land aboard carriers*. 2005.
29. Système PDG®, <http://www.uqtr.ca/~larepel/Larepe/pdg/index.php>, 1999.

30. Turban, E. et J.E. Aronson, *Decision Support Systems and Intelligent Systems*. 2001: Prentice Hall.
31. Université du Québec à Trois-Rivières, <http://www.uqtr.ca/>.
32. Visual Mining Inc, <http://www.visualmining.com/>.
33. Yasin, M.M., *The Theory and Practice of Benchmarking: Then and Now*. Benchmarking: An International Journal, 2002. 9(3): pp.217-243.